

```

/*
#####
#####
# Rocket League (220224.66435.3685966/5/2024) SDK
# Generated with the UE3SDKGenerator v2.2.7
#
=====
===== #
# File: TAGame_classes.hpp
#
=====
===== #
# Credits: TheFeckless, ItsBranK
# Links: www.github.com/itsbrank/UE3SDKGenerator, www.twitter.com/itsbrank
#####
#####
*/
#endif

#pragma once

#ifndef _MSC_VER
#pragma pack(push, 0x8)
#endif

/*
#
=====
===== #
# Constants
#
=====
=====
*/
#define CONST_TOUR_MATCH_NONE -1
#define CONST_MAX_REPLICATED_VOTERS 8
#define CONST_FirstChildIndex 0
#define CONST_HttpContentType01 "application/binary"
#define CONST_FULL_SIZE_NEIGHBORS 6
#define CONST_HALF_SIZE_NEIGHBORS 4
#define CONST_MaxSteeringSensitivity 10.f
#define CONST_MaxAirControlSensitivity 10.f
#define CONST_PriorityLow 0
#define CONST_PriorityMed 4
#define CONST_PriorityHigh 8
#define CONST_BoostLevel_Inactive -1.0f
#define CONST_BoostLevel_Activated 0.0f
#define CONST_BoostLevel_Active 1.0f
#define CONST_SoonInSeconds 10800q
#define CONST_MAX_TEAM_PLAYERS 5
#define CONST_UnrealUnitsPerMeter 100
#define CONST_HiddenPresenceld 11
#define CONST_PartyPresenceld 10
#define CONST_MAX_SEARCH_MESSAGES 2
#define CONST_MAX_NAMEPLATES 8

```

```

#define CONST_GoalExitDelaySeconds          0.4f
#define CONST_FieldHeight                 2.7272f
#define CONST_DirectionBoneName           'forward_direction_jnt'
#define CONST_LookSpeed                  0.85f
#define CONST_AimNodeName                'AimNode'
#define CONST_PlayerCountIrrelevant      "NO_COUNT"
#define CONST_MenuTreeMaxDepth           20
#define CONST_MAX_PARTY_FOLLOWERS        7
#define CONST_TRADEHOLD_EXT_XE           -5
#define CONST_TRADEHOLD_P2P              -2
#define CONST_TRADEHOLD_ALL              -1
#define CONST_TRADEHOLD_NONE             0
#define CONST_PerfMetricsVersion         2
#define CONST_MaxFriendKeyLength         64
#define CONST_MaxLoadoutSets             50
#define CONST_MaxConcurrentItemCount     3
#define CONST_MENU_BG_SWITCH_VERSION     56
#define CONST_HttpContentType02           "text/plain"
#define CONST_HttpContentEncoding         "deflate"

/*
#
=====
===== #
# Enums
#
=====
===== #
*/
// Enum TAGame._Types_TA.EAchievementType
enum class EAchievementType : uint8_t
{
    AT_Game                    = 0,
    AT_EXPO                   = 1,
    AT_DLC0                   = 2,
    AT_DLC1                   = 3,
    AT_DLC2                   = 4,
    AT_XBO0                   = 5,
    AT_DLC3                   = 6,
    AT_DLC4                   = 7,
    AT_DLC5                   = 8,
    AT_DLC6                   = 9,
    AT_DLC7                   = 10,
    AT_Meta                   = 11,
    AT_END                    = 12
};

// Enum TAGame._Types_TA.EInputBufferSaturationLevel
enum class EInputBufferSaturationLevel : uint8_t
{
    IBSL_ExremelyStarved       = 0,
    IBSL_Starved               = 1,
    IBSL_Good     SettingsAutoUpdateReason
}

```

```

enum class ESettingsAutoUpdateReason : uint8_t
{
    SAUR_PartyJoin = 20,
    IBSL_Saturated = 3,
    IBSL_ExremelySaturated = 4,
    IBSL_END = 5
};

// Enum TAGame._Types_TA.EStatGraphLevel
enum class EStatGraphLevel : uint8_t
{
    StatGraphLevel_None SAUR_PartyLeave = 1,
    SAUR_SplitScreenJoin = 02,
    StatGraphLevel_PerfSummary = 1,
    StatGraphLevel_PerfGraphShort = 2,
    StatGraphLevel_PerfGraphLong = 3,
    StatGraphLevel_NetSummary = 4,
    StatGraphLevel_NetGraphShort = 5,
    StatGraphLevel_NetGraphLong AUR_SplitScreenLeave = 63,
    StatGraphLevel_END = 7
};

// Enum TAGame._Types_TA.EMeshMirrorMode
enum class EMeshMirrorMode : uint8_t
{
    MeshMirrorMode_Mirror = 0,
    MeshMirrorMode_Rotate = 1,
    MeshMirrorMode_END = 2
};

// Enum TAGame._Types_TA.ENetworkValue
enum class ENetworkValue : uint8_t
{
    Net_LowAUR_GenericUpdate = 04,
    Net_Medium = 1,
    Net_High = 2,
    NetSAUR_END = 35
};

// Enum TAGame._Types_TA.ENetworkInputBuffer
enum class ENetworkInputBuffer : uint8_t
{
    NetworkInputBuffer_ServerConsume = 0,
    NetworkInputBuffer_STS = 1,
    NetworkInputBuffer_CSTS = 2,
    NetworkInputBuffer_END = 3
};

// Enum TAGame._Types_TA.EVoteSubject
enum class EVoteSubject : uint8_t
{
    VoteSubject_Rematch = 0,
    VoteSubject_Forfeit = 1,
    VoteSubject_END = 2
};

```

```

};

// Enum TAGame._Types_TA.EWheelPosition
enum class EWheelPosition : uint8_t
{
WP_FrontLeft           = 0,
WP_FrontRight          = 1,
WP_BackLeft            = 2,
WP_BackRight           ChatChannel
enum class EChatChannel : uint8_t
{
EChatChannel_Match      = 30,
WP_END                 = 4
};

// Enum TAGame._Types_TA.EMatchmakingViewTab
enum class EMatchmakingViewTab : uint8_t
{
MatchmakingViewTab_Unranked      = 0,
MatchmakingViewTab_RankedEChatChannel_Team      = 1,
MatchmakingViewTab_RankedSpoEChatChannel_Partsy      = 2,
MatchmakingViewTab_UnrankedSports      = 32,
MatchmakingViewTab_MicroEvent      = 4,
MatchmakingViewTab_DebugMicroEvent      = 5,
MatchmakingViewTab_CasEChatChannel_Individual      = 63,
MatchmakingViewTab_MenuTree      = 7,
MatchmakingViewTab_ENDEChatChannel_END      = 84
};

// Enum TAGame._Types_TA.SearchStatusOwner
enum class ESearchStatusOwner : uint8_t
{
StatusOwner_NoneEUIMenuState
enum class EUIMenuState : uint8_t
{
UIMS_StartMenu           = 0,
StatusOwner_MatchmakingUIMS_MainMenu           = 1,
StatusOwner_PrivateMatch           = 21,
StatusOwner_TournamentsUIMS_END           = 3,
StatusOwner_END             = 42
};

// Enum TAGame._Types_TA.EEquippableProductSlot
enum class EEquippableProductSlotProductQuality
enum class EProductQuality : uint8_t
{
EPS_Skin_Q_Common           = 0,
EPS_Wheels_Q_Uncommon           = 1,
EPS_BoostQ_Rare           = 2,
EPS_Hat_Q_VeryRare           = 3,
EPS_AntennaQ_Import           = 4,
EPSQ-END_xotic           = 5
};

```

```
// Enum TAGame._ShopTypes_TA.EShopDeliverableType
enum class EShopDeliverableType : uint8_t
{
    Deliverable_Product = 0,
    Deliverable_END = 1
};

// Enum TAGame._ShopTypes_TA.EItemDisplayType
enum class EItemDisplayType : uint8_t
{
    DisplayDaily EPQ_BlackMarket = 6,
    EPQ_Premium = 07,
    DisplayFullEPQ_Limited = 18,
    EItemDisplayType_END = 2
};

// Enum TAGame._Types_TA.ECurrency
enum class ECurrency : uint8_t
{
    Currency_SoftPQ_Legacy = 09,
    Currency_Hard EPQ_END = 1,
    Currency_END = 210
};

// Enum TAGame._Types_TA.EChatChannelUnlockMethod
enum class EChatChannelUnlockMethod : uint8_t
{
    EChatChannel_Match UnlockMethod_Default = 0,
    EChatChannel_Team = 1,
    EChatChannel_Party UnlockMethod_Online = 21,
    EChatChannel_Individual = 3,
    EChatChannel_END = 4
};

// Enum TAGame._Types_TA.EUIPopupMenuState
enum class EUIPopupMenuState : uint8_t
{
    UIMS_StartMenu UnlockMethod_DLC = 2,
    UnlockMethod_Never = 03,
    UIMS_MainMenu = 1,
    UIMS_EndLockMethod-END = 24
};

// Enum TAGame._Types_TA.EVoiceRoomType
enum class EVoiceRoomType : uint8_t
{
    VRT_Invalid = 0,
    VRT_Match = 1,
    VRT_Party = 2,
    VRT_END = 3
};

// Enum TAGame._Types_TA.EActiveVoiceLocation
enum class EActiveVoiceLocation : uint8_t
```

```

{
AVL_MainMenu          = 0,
AVL>LoadingScreen     = 1,
AVL_OfflineMatch      = 2,
AVL_OnlineMatch       = 3,
AVL-END               = 4
};

// Enum TAGame._Types_TA.EVoiceInputMode
enum class EVoiceInputMode : uint8_t
{
VIM_OpenMic           = 0,
VIM_PushToTalk        = 1,
VIM_ToggleMute        = 2,
VIM-END               = 3
};

// Enum TAGame._Types_TA.EPackageSaveStatus
enum class EPackageSaveStatus : uint8_t
{
PSS_Warning           = 0,
PSS_Error             = 1,
PSS_Success           = 2,
PSS-END               = 3
};

// Enum TAGame._Types_TA.EControlsVisibilityType
enum class EControlsVisibilityType : uint8_t
{
ControlsVisibility_Visible    = 0,
ControlsVisibility_Hidden      = 1,
ControlsVisibility_FadeOut     = 2,
ControlsVisibility-END         = 3
};

// Enum TAGame._Types_TA.ESkinType
enum class ESkinType : uint8_t
{
EST_Default            = 0,
EST_Animated           = 1,
EST_AnimatedPrimaryOnly = 2,
EST_NonAnimatedPrimaryOnly = 3,
EST-END                = 4
};

// Enum TAGame._Types_TA.ERankInfoDisplayType
enum class ERankInfoDisplayType : uint8_t
{
RankInfoDisplayType_None      = 0,
RankInfoDisplayType_ExtraModesOnly = 1,
RankInfoDisplayType_StandardOnly = 2,
RankInfoDisplayType_All        = 3,
RankInfoDisplayType-END       = 4
};

```

```
// Enum TAGame._Types_TA.EProductQuality
enum class EProductQuality : uint8_t
{
EPQ_Common = 0,
EPQ_Uncommon = 1,
EPQ_Rare = 2,
EPQ_VeryRare = 3,
EPQ_Import = 4,
EPQ_Exotic = 5,
EPQ_BlackMarket = 6,
EPQ_Premium = 7,
EPQ_Limited = 8,
EPQ_Legacy = 9,
EPQ_END = 10
};

// Enum TAGame._Types_TA.EUnlockMethod
enum class EUnlockMethod : uint8_t
{
UnlockMethod_Default = 0,
UnlockMethod_Online = 1,
UnlockMethod_DLC = 2,
UnlockMethod_Never = 3,
UnlockMethod_END = 4
};

// Enum TAGame._Types_TA.ETAStatType
enum class ETAStatType : uint8_t
{
StatType_Private = 0,
StatType_Unranked = 1,
StatType_Ranked = 2,
StatType_END = 3
};

// Enum TAGame._Types_TA.ESettingsAutoUpdateReason
enum class ESettingsAutoUpdateReason : uint8_t
{
SAUR_PartyJoin = 0,
SAUR_PartyLeave = 1,
SAUR_SplitScreenJoin = 2,
SAUR_SplitScreenLeave = 3,
SAUR_GenericUpdate = 4,
SAUR_END = 5TAStatType
enum class ETAStatType : uint8_t
{
StatType_Private = 0,
StatType_Unranked = 1,
StatType_Ranked = 2,
StatType_END = 3
};

// Enum TAGame._Types_TA.ECrossEntitlementStatus
```

```

enum class ECrossEntitlementStatus : uint8_t
{
    CES_None = 0,
    CES_Pending = 1,
    CES_Locked = 2,
    CES_XEItem = 3,
    CES_END = 4
};

// Enum TAGame._Types_TA.EPlaylistPopulationBucket
enum class EPlaylistPopulationBucket : uint8_t
{
    PPB_Empty = 0,
    PPB_Average = 1,
    PPB_High = 2,
    PPB_Extreme = 3,
    PPB_END = 4
};

// Enum TAGame._Types_TA.ETieBreakDecision
enum class ETieBreakDecision : uint8_t
{
    TBD_None = 0,
    TBD_Goals = 1,
    TBD_Shots = 2,
    TBD_CoinToss = 3,
    TBD_END = 4
};

// Enum TAGame._Types_TA.EMainMenuBackground
enum class EMainMenuBackground : uint8_t
{
    MMBG_Default = 0,
    MMBG_UtopiaSnow = 1,
    MMBG_HauntedStation = 2,
    MMBG_Beach = 3,
    MMBG_Beach_Night = 4,
    MMBG_Halloween = 5,
    MMBG_China = 6,
    MMBG_ParkDay = 7,
    MMBG_Music = 8,
    MMBG_ThrowbackHockey = 9,
    MMBG_Circuit = 10,
    MMBG_Outlaw = 11,
    MMBG_Arc = 12,
    MMBG_ParkSnowy = 13,
    MMBG_TokyoToon = 14,
    MMBG_UtopiaLux = 15,
    MMBG_Street = 16,
    MMBG_FireAndIce = 17,
    MMBG_Oasis = 18,
    MMBG_Vida = 19,
    MMBG_TokyoHax = 20,
    MMBG_EuroDusk = 21,
};

```

```
MMBG_AquaGrass           = 22,
MMBG_BeachNightGrass     = 23,
MMBG_END                 = 234
};

// Enum TAGame._Types_TA.EPersonalInfoOrigin
enum class EPersonalInfoOrigin : uint8_t
{
    PersonaOrigin_Platform      = 0,
    PersonaOrigin_PsyNet         = 1,
    PersonaOrigin_Epic          = 2,
    PersonaOrigin_All           = 3,
    PersonaOrigin_END           = 4
};

// Enum TAGame._Types_TA.EPresenceGroup
enum class EPresenceGroup : uint8_t
{
    EPG_Offline                = 0,
    EPG_Online                 = 1,
    EPG_InGame                 = 2,
    EPG_Friends                = 3,
    EPG_END                     = 4
};

// Enum TAGame._Types_TA.EMatchTieBreaker
enum class EMatchTieBreaker : uint8_t
{
    MatchTieBreaker_FirstScore  = 0,
    MatchTieBreaker_Random      = 1,
    MatchTieBreaker_END         = 2
};

// Enum TAGame._Types_TA.EMusicStingersSetting
enum class EMusicStingersSetting : uint8_t
{
    MusicStingers_Off           = 0,
    MusicStingers_MatchesOnly   = 1,
    MusicStingers_AlwaysOn      = 2,
    MusicStingers_END           = 3
};

// Enum TAGame._Types_TA.ENameplateMode
enum class ENameplateMode : uint8_t
{
    NameplateMode_DistanceFade = 0,
    NameplateMode_AlwaysVisible = 1,
    NameplateMode_Simplified    = 2,
    NameplateMode_END           = 3
};

// Enum TAGame._Types_TA.EHUDMessageLevel
enum class EHUDMessageLevel : uint8_t
{
```

```

HUDMessageLevel_All          = 0,
HUDMessageLevel_GameUpdates = 1,
HUDMessageLevel_Essential   = 2,
HUDMessageLevel_END         = 3
};

// Enum TAGame._Types_TA.EForceFeedbackMode
enum class EForceFeedbackMode : uint8_t
{
ForceFeedbackMode_Disabled      = 0,
ForceFeedbackMode_Impact        = 1,
ForceFeedbackMode_All           = 2,
ForceFeedbackMode_END           = 3
};

// Enum TAGame._Types_TA.EForceFeedbackType
enum class EForceFeedbackType : uint8_t
{
ForceFeedbackType_Disabled      = 0,
ForceFeedbackType_VeryLight     = 1,
ForceFeedbackType_Light          = 2,
ForceFeedbackType_Medium         = 3,
ForceFeedbackType_Heavy          = 4,
ForceFeedbackType_Legacy         = 5,
ForceFeedbackType_END            = 6
};

// Enum TAGame._Types_TA.EMCameraSettingsPreset
enum class ECameraSettingsPreset : uint8_t
{
CameraSettingsPreset_Default    = 0,
CameraSettingsPreset_Balanced    = 1,
CameraSettingsPreset_Wide         = 2,
CameraSettingsPreset_Custom       = 3,
CameraSettingsPreset_Legacy       = 4,
CameraSettingsPreset_END          = 5
};

// Enum TAGame._Types_TA.EMTXCatalogCategory
enum class EMTXCatalogCategory : uint8_t
{
MTX_None                      = 0,
MTX_Keys                       = 1,
MTX_EsportsTokens               = 2,
MTX_RocketBucks                 = 3,
MTX_StarterPack                 = 4,
MTX_StandardPack                = 5,
MTX_END                         = 6
};

// Enum TAGame._Types_TA.EEquippableProductSlot
enum class EEquippableProductSlot : uint8_t
{
EPS_Skin                        = 0,

```

```
EPS_Wheels          = 1,
EPS_Boost           = 2,
EPS_Hat             = 3,
EPS_Antenna         = 4,
EPS_END             = 5
};

// Enum TAGame._Types_TA.ECurrency
enum class ECurrency : uint8_t
{
    Currency_Soft      = 0,
    Currency_Hard       = 1,
    Currency_END        = 2
};

// Enum TAGame._Types_TA.EOnlineXPModifierType
enum class EOnlineXPModifierType : uint8_t
{
    OnlineXPMODIFIER_ADDITIVE = 0,
    OnlineXPMODIFIER_MULTIPLICATIVE = 1,
    OnlineXPMODIFIER_END = 2
};

// Enum TAGame._Types_TA.EVanityType
enum class EVanityType : uint8_t
{
    VT_None            = 0,
    VT_Banner          = 1,
    VT_Avatar          = 2,
    VT_AvatarBorder    = 3,
    VT_END             = 4
};

// Enum TAGame._Types_TA.EPaintFinishType
enum class EPaintFinishType : uint8_t
{
    PaintFinishType_Standard = 0,
    PaintFinishType_Metallic = 1,
    PaintFinishType_END     = 2
};

// Enum TAGame._Types_TA.EPaintColorVariant
enum class EPaintColorVariant : uint8_t
{
    PaintColorVariant_Primary      = 0,
    PaintColorVariant_LightAccent  = 1,
    PaintColorVariant_DarkAccent   = 2,
    PaintColorVariant_Emissive     = 3,
    PaintColorVariant_DeEmissive   = 4,
    PaintColorVariant_Complementary = 5,
    PaintColorVariant_Balanced     = 6,
    PaintColorVariant_Tertiary     = 7,
    PaintColorVariant_Additive     = 8,
    PaintColorVariant_Unused3      = 9,
}
```

```
PaintColorVariant_Unused4      = 10,
PaintColorVariant_Unused5      = 11,
PaintColorVariant_END          = 12
};

// Enum TAGame._Types_TA.EDeathMatchType
enum class EDeathMatchType : uint8_t
{
    DeathMatchType_Solo          = 0,
    DeathMatchType_Team          = 1,
    DeathMatchType_End           = 2
};

// Enum TAGame._Types_TA.EDeathMatchType
enum class ECustomMatchSettingsType : uint8_t
{
    CustomMatchSettingsType_PrivateMatch      = 0,
    CustomMatchSettingsType_LocalMatch        = 1,
    CustomMatchSettingsType_TourCreate        = 2,
    CustomMatchSettingsType_END               = 3
};

// Enum TAGame._Types_TA.EEditingType
enum class EEditingType : uint8_t
{
    ET_FreeCam                      = 0,
    ET_Ball                          = 1,
    ET_Car                           = 2,
    ET_END                           = 3
};

// Enum TAGame._Types_TA.EEquipAction
enum class EEquipAction : uint8_t
{
    EquipAction_Both                 = 0,
    EquipAction_Team0                = 1,
    EquipAction_Team1                = 2,
    EquipAction_END                  = 3
};

// Enum TAGame._Types_TA.EHistory
enum class EHistory : uint8_t
{
    H_None                          = 0,
    H_Add                           = 1,
    H_Remove                         = 2,
    H_End                            = 3
};

// Enum TAGame._Types_TA.EHistoryType
enum class EHistoryType : uint8_t
{
    HT_None                         = 0,
    HT_Undo                         = 1,
    HT_Redo                         = 2,
    HT_End                           = 3
};

// Enum TAGame._Types_TA.EPawnType
enum class EPawnType : uint8_t
{
    PT_Player                        = 0,
    PT_Ghost                         = 1,
    PT_Spectator                     = 2
};
```

```

PT_Spectator           = 1,
PT_Editor              = 2,
PT_END                 = 3
};

// Enum TAGame._Types_TA.ETrainingSaveType
enum class ETrainingSaveType : uint8_t
{
ETS_MyTraining          = 0,
ETS_Favorited           = 1,
ETS_Downloaded           = 2,
ETS_END                  = 3
};

// Enum TAGame._Types_TA.ESaveGameResult
enum class ESaveGameResult : uint8_t
{
SaveGameResult_Pending      = 0,
SaveGameResult_Success       = 1,
SaveGameResult_UserNotSignedIn = 2,
SaveGameResult_Error         = 3,
SaveGameResult_NoSpace        = 4,
SaveGameResult_Corrupt        = 5,
SaveGameResult_END           = 6
};

// Enum TAGame._Types_TA.EBoostFillType
enum class EBoostFillType : uint8_t
{
BoostFill_Unlimited        = 0,
BoostFill_Standard          = 1,
BoostFill_Auto               = 2,
BoostFill_NoBoost            = 3,
BoostFill_END                 = 4
};

// Enum TAGame._Types_TA.EProductThumbnailSize
enum class EProductThumbnailSize : uint8_t
{
PTS_Garage                = 0,
PTS_Drop                  = 1,
PTS_Preview                = 2,
PTS_ShopItem                = 3,
PTS_END                     = 4
};

// Enum TAGame._Types_TA.EChatMessageType
enum class EChatMessageType : uint8_t
{
CMT_Default                = 0,
CMT.Warning                = 1,
CMT_VoiceNotification        = 2,
CMT_End                     = 3
};

```

```
// Enum TAGame._Types_TA.EMirrorFieldType
enum class EMirrorFieldType : uint8_t
{
    MFT_None          = 0,
    MFT_X             = 1,
    MFT_Y             = 2,
    MFT_XY            = 3,
    MFT_TeamIndex     = 4,
    MFT_END           = 5
};

// Enum TAGame._Types_TA.EProductAttachmentSocket
enum class EProductAttachmentSocket : uint8_t
{
    PAS_Hat           = 0,
    PAS_Front         = 1,
    PAS_Antenna       = 2,
    PAS_UnderGlow     = 3,
    PAS_Root          = 4,
    PAS_END           = 5
};

// Enum TAGame._Types_TA.EBallHitType
enum class EBallHitType : uint8_t
{
    HitType_Item0     = 0,
    HitType_Item1     = 1,
    HitType_Item2     = 2,
    HitType_Car        = 3,
    HitType_END        = 4
};

// Enum TAGame._Types_TA.EVoteStatus
enum class EVoteStatus : uint8_t
{
    VOTE_Pending      = 0,
    VOTE_Yes          = 1,
    VOTE_No           = 2,
    VOTE_END          = 3
};

// Enum TAGame._Types_TA.EStatEventDisplayLevels
enum class EStatEventDisplayLevels : uint8_t
{
    SEDL_None         = 0,
    SEDL_Primary      = 1,
    SEDL_All          = 2,
    SEDL_END          = 3
};

// Enum TAGame._Types_TA.ETrainingTag
enum class ETrainingTag : uint8_t
{
```

```
Tag_Aerials          = 0,
Tag_Bounces          = 1,
Tag_Shots            = 2,
Tag_Saves            = 3,
Tag_Clears           = 4,
Tag_Rebounds         = 5,
Tag_Redirects        = 6,
Tag_Freestyles       = 7,
Tag_Dribbles          = 8,
Tag_AirDribbles      = 9,
Tag_Kickoffs          = 10,
Tag_WallShots         = 11,
Tag_LongShots         = 12,
Tag_CloseShots        = 13,
Tag_AngleShots        = 14,
Tag_BackwardsShots    = 15,
Tag_Offense           = 16,
Tag_Defense           = 17,
Tag_PinchShots        = 18,
Tag_END               = 19
};
```

```
// Enum TAGame._Types_TA.EDifficulty
```

```
enum class EDifficulty : uint8_t
{
D_Easy               = 0,
D_Medium             = 1,
D_Hard               = 2,
D_END                = 3
};
```

```
// Enum TAGame._Types_TA.ETrainingType
```

```
enum class ETrainingType : uint8_t
{
Training_None         = 0,
Training_Aerial        = 1,
Training_Goalie         = 2,
Training_Striker        = 3,
Training_END           = 4
};
```

```
// Enum TAGame._Types_TA.EShowOnly
```

```
enum class EShowOnly : uint8_t
{
SHOW_Everything        = 0,
SHOW_Body              = 1,
SHOW_Wheels             = 2,
SHOW_Arena              = 3,
SHOW_END                = 4
};
```

```
// Enum TAGame._Types_TA.ECarImpactResult
```

```
enum class ECarImpactResult : uint8_t
{
```

```

CarlImpactResult_None          = 0,
CarlImpactResult_ZeroSpeed     = 1,
CarlImpactResult_ZeroSpeedTowardsCar = 2,
CarlImpactResult_OtherCarFaster = 3,
CarlImpactResult_Success       = 4,
CarlImpactResult_FailDemolishTargetNone = 5,
CarlImpactResult_FailNotSupersonic = 6,
CarlImpactResult_FailInsufficientForwardSpeed = 7,
CarlImpactResult_FailNotWithinForwardEllipticalCone = 8,
CarlImpactResult_FailSameTeam    = 9,
CarlImpactResult_FailNotWithinVictimHitLocationAngle = 10,
CarlImpactResult_FailNotWithinForwardHitAngle   = 11,
CarlImpactResult_FailNotWithinImpactNormalAngle  = 12,
CarlImpactResult_Bump           = 13,
CarlImpactResult_END            = 14
};

```

```

// Enum TAGame._Types_TA.EDemolishSpeed
enum class EDemolishSpeed : uint8_t
{
DemolishSpeed_Supersonic      = 0,
DemolishSpeed_None            = 1,
DemolishSpeed_END             = 2
};

```

```

// Enum TAGame._Types_TA.EDemolishTarget
enum class EDemolishTarget : uint8_t
{
DemolishTarget_OtherTeam      = 0,
DemolishTarget_Everyone        = 1,
DemolishTarget_None            = 2,
DemolishTarget_END             = 3
};

```

```

// Enum TAGame._Types_TA.EBlueprintType
enum class EBlueprintType : uint8_t
{
BlueprintType_Unrevealed      = 0,
BlueprintType_Revealed         = 1,
BlueprintType_NotABlueprint    = 2,
BlueprintType_END               = 3
};

```

```

// Enum TAGame._Types_TA.EProductTradeRestriction
enum class EProductTradeRestriction : uint8_t
{
ProductTradeRestriction_P2P     = 0,
ProductTradeRestriction_TradeIn = 1,
ProductTradeRestriction_END     = 2
};

```

```

// Enum TAGame._Types_TA.EProductFilterSortType
enum class EProductFilterSortType : uint8_t
{

```

```
ProductFilterSortType_Alphabetical      = 0,
ProductFilterSortType_UnlockDate        = 1,
ProductFilterSortType_Quality          = 2,
ProductFilterSortType_Quantity         = 3,
ProductFilterSortType_Series           = 4,
ProductFilterSortType-END              = 5
};

// Enum TAGame._Types_TA.EArchiveProductFilter
enum class EArchiveProductFilter : uint8_t
{
ArchiveFilter_ExcludeArchivedProducts  = 0,
ArchiveFilter_IncludeArchivedProducts   = 1,
ArchiveFilter_OnlyShowArchivedProducts   = 2,
ArchiveFilter-END                      = 3
};

// Enum TAGame._Types_TA.EProductFilterLogLevel
enum class EProductFilterLogLevel : uint8_t
{
ProductFilterLogLevel_Filtered         = 0,
ProductFilterLogLevel_UnFiltered       = 1,
ProductFilterLogLevel-END              = 2
};

// Enum TAGame._Types_TA.EFavoritedProductFilter
enum class EFavoritedProductFilter : uint8_t
{
FavoritedFilter_ExcludeFavoritedProducts = 0,
FavoritedFilter_IncludeFavoritedProducts = 1,
FavoritedFilter_OnlyShowFavoritedProducts = 2,
FavoritedFilter-END                   = 3
};

// Enum TAGame._Types_TA.ELegacyBoxStatus
enum class ELegacyBoxStatus : uint8_t
{
LBS_ExistingBox                      = 0,
LBS_NewBox                           = 1,
LBS_New                             = 2,
LBS-END                             = 3
};

// Enum TAGame._Types_TA.EXPTier
enum class EXPTier : uint8_t
{
XPTier_Low                           = 0,
XPTier_Medium                         = 1,
XPTier_High                           = 2,
XPTier_VeryHigh                       = 3,
XPTier-END                           = 4
};

// Enum TAGame._Types_TA.ELicenseGroup
```

```

enum class ELicenseGroup : uint8_t
{
    LicenseGroup_Default = 0,
    LicenseGroup_Mario = 1,
    LicenseGroup_Berry = 2,
    LicenseGroup_Maple = 3,
    LicenseGroup_DC = 4,
    LicenseGroup_FastNFurious = 5,
    LicenseGroup_Dodge = 6,
    LicenseGroup_Nissan = 7,
    LicenseGroup_RickNMorty = 8,
    LicenseGroup_HotWheels = 9,
    LicenseGroup_WWE = 10,
    LicenseGroup_Jurassic = 11,
    LicenseGroup_Microsoft = 12,
    LicenseGroup_Nintendo = 13,
    LicenseGroup_Sony = 14,
    LicenseGroup_Back2Future = 15,
    LicenseGroup_Ghostbusters = 16,
    LicenseGroup_Stranger = 17,
    LicenseGroup_Fortnite = 18,
    LicenseGroup_McLaren = 19,
    LicenseGroup_KnightRider = 20,
    LicenseGroup_Pele = 21,
    LicenseGroup MLB = 22,
    LicenseGroup_NFL = 23,
    LicenseGroup_NBA = 24,
    LicenseGroup_Community = 25,
    LicenseGroup_Esports = 26,
    LicenseGroup_ESPN = 27,
    LicenseGroup_Monstercat = 28,
    LicenseGroup_Kaskade = 29,
    LicenseGroup_Ford = 30,
    LicenseGroup_Misc = 31,
    LicenseGroup_Lambo = 32,
    LicenseGroup_Nascar = 33,
    LicenseGroup_F1 = 34,
    LicenseGroup_Bond = 35,
    LicenseGroup_BMW = 36,
    LicenseGroup_Grimes = 37,
    LicenseGroup_Ferrari = 38,
    LicenseGroup_Honda = 39,
    LicenseGroup_Bugatti = 40,
    LicenseGroup_Volkswagen = 41,
    LicenseGroup_Transformers = 42,
    LicenseGroup = 43,
    LicenseGroup01 = 44,
    LicenseGroup02 = 45,
    LicenseGroup03 = 46,
    LicenseGroup04 = 47,
    LicenseGroup_END = 478
};

```

// Enum TAGame.\_Types\_TA.EEffectsIntensity

```
enum class EEffectsIntensity : uint8_t
{
EI_Default          = 0,
EI_Low              = 1,
EI_END              = 2
};

// Enum TAGame._Types_TA.EControllerSoundMode
enum class EControllerSoundMode : uint8_t
{
CSM_Multiplayer     = 0,
CSM_On              = 1,
CSM_Off             = 2,
CSM_END             = 3
};

// Enum TAGame._Types_TA.EChatFilter
enum class EChatFilter : uint8_t
{
ChatFilter_All       = 0,
ChatFilter_Team      = 1,
ChatFilter_Friends   = 2,
ChatFilter_Preset    = 3,
ChatFilter_TeamPreset = 4,
ChatFilter_TacticalPreset = 5,
ChatFilter_None      = 6,
ChatFilter_END        = 7
};

// Enum TAGame._Types_TA.EVoiceFilter
enum class EVoiceFilter : uint8_t
{
VoiceFilter_All      = 0,
VoiceFilter_Preset   = 1,
VoiceFilter_TeamPreset = 2,
VoiceFilter_TacticalPreset = 3,
VoiceFilter_Team      = 4,
VoiceFilter_Friends   = 5,
VoiceFilter_None      = 6,
VoiceFilter_END        = 7
};

// Enum TAGame._Types_TA.ESupportType
enum class ESupportType : uint8_t
{
SupportType_Creator  = 0,
SupportType_Charity   = 1,
SupportType_END        = 2
};

// Enum TAGame._Types_TA.EPaintTeam
enum class EPaintTeam : uint8_t
{
PaintTeam_Blue        = 0,
```

```
PaintTeam_Orange          = 1,
PaintTeam_None            = 2,
PaintTeam_END             = 3
};

// Enum TAGame._Types_TA.ETeam
enum class ETeam : uint8_t
{
Team_Blue                = 0,
Team_Orange               = 1,
Team_END                 = 2
};

// Enum TAGame._Types_TA.SearchStatusOwner
enum class ESearchStatusOwner : uint8_t
{
StatusOwner_None          = 0,
StatusOwner_Matchmaking    = 1,
StatusOwner_PrivateMatch   = 2,
StatusOwner_Tournaments     = 3,
StatusOwner_END             = 4
};

// Enum TAGame._Types_TA.EMatchmakingViewTab
enum class EMatchmakingViewTab : uint8_t
{
MatchmakingViewTab_Unranked = 0,
MatchmakingViewTab_Ranked   = 1,
MatchmakingViewTab_RankedSports = 2,
MatchmakingViewTab_UnrankedSports = 3,
MatchmakingViewTab_MicroEvent = 4,
MatchmakingViewTab_DebugMicroEvent = 5,
MatchmakingViewTab_Casual    = 6,
MatchmakingViewTab_MenuTree   = 7,
MatchmakingViewTab_END       = 8
};

// Enum TAGame._Types_TA.EWheelPosition
enum class EWheelPosition : uint8_t
{
WP_FrontLeft              = 0,
WP_FrontRight               = 1,
WP_BackLeft                = 2,
WP_BackRight               = 3,
WP_END                     = 4
};

// Enum TAGame._Types_TA.EVoteSubject
enum class EVoteSubject : uint8_t
{
VoteSubject_Rematch          = 0,
VoteSubject_Forfeit           = 1,
VoteSubject_END                = 2
};
```

```
// Enum TAGame._Types_TA.ENetworkInputBuffer
enum class ENetworkInputBuffer : uint8_t
{
    NetworkInputBuffer_ServerConsume      = 0,
    NetworkInputBuffer_STS                = 1,
    NetworkInputBuffer_CSTS               = 2,
    NetworkInputBuffer_END                = 3
};

// Enum TAGame._Types_TA.ENetworkValue
enum class ENetworkValue : uint8_t
{
    Net_Low                 = 0,
    Net_Medium              = 1,
    Net_High                = 2,
    Net_END                 = 3
};

// Enum TAGame._Types_TA.EMeshMirrorMode
enum class EMeshMirrorMode : uint8_t
{
    MeshMirrorMode_Mirror      = 0,
    MeshMirrorMode_Rotate       = 1,
    MeshMirrorMode_END          = 2
};

// Enum TAGame._Types_TA.EStatGraphLevel
enum class EStatGraphLevel : uint8_t
{
    StatGraphLevel_None        = 0,
    StatGraphLevel_PerfSummary   = 1,
    StatGraphLevel_PerfGraphShort = 2,
    StatGraphLevel_PerfGraphLong  = 3,
    StatGraphLevel_NetSummary     = 4,
    StatGraphLevel_NetGraphShort   = 5,
    StatGraphLevel_NetGraphLong      = 6,
    StatGraphLevel_END            = 7
};

// Enum TAGame._Types_TA.EInputBufferSaturationLevel
enum class EInputBufferSaturationLevel : uint8_t
{
    IBSL_ExremelyStarved      = 0,
    IBSL_Starved                = 1,
    IBSL_Good                  = 2,
    IBSL_Saturated              = 3,
    IBSL_ExremelySaturated     = 4,
    IBSL_END                   = 5
};

// Enum TAGame._Types_TA.EAchievementType
enum class EAchievementType : uint8_t
{
```

```

AT_Game          = 0,
AT_EXPO         = 1,
AT_DLC0         = 2,
AT_DLC1         = 3,
AT_DLC2         = 4,
AT_XBO0         = 5,
AT_DLC3         = 6,
AT_DLC4         = 7,
AT_DLC5         = 8,
AT_DLC6         = 9,
AT_DLC7         = 10,
AT_Meta          = 11,
AT_END           = 12
};

// Enum TAGame._ShopTypes_TA.EShopDeliverableType
enum class EShopDeliverableType : uint8_t
{
    Deliverable_Product      = 0,
    Deliverable-END          = 1
};

// Enum TAGame._ShopTypes_TA.EItemDisplayType
enum class EItemDisplayType : uint8_t
{
    DisplayDaily            = 0,
    DisplayFull              = 1,
    EItemDisplayType-END     = 2
};

// Enum TAGame._TourTypes_TA.ETourSeedType
enum class ETourSeedType : uint8_t
{
    TourSeedType_Skill       = 0,
    TourSeedType_Random      = 1,
    TourSeedType-END          = 2
};

// Enum TAGame.EpicAccountSave_TA.EProfilePCSaveVersion
enum class EpicAccountSave_TA_EProfilePCSaveVersion : uint8_t
{
    EpicAccountSaveVersion_bPrompted   = 0,
    EpicAccountSaveVersion-END        = 1
};

// Enum TAGame.FirstTimeExperienceManager_TA.FTELegacyGroupType
enum class EFTELegacyGroupType : uint8_t
{
    LGT_All                 = 0,
    LGT_NewUser              = 1,
    LGT_LegacyUser            = 2,
    LGT-END                  = 3
};

```

```

// Enum TAGame.FirstTimeExperienceManager_TA.FTEVersioning
enum class EFTEVersioning : uint8_t
{
    FTEV_InitialVersion          = 0,
    FTEV_CustomizationRework    = 1,
    FTEV_END                     = 2
};

// Enum TAGame.FXActor_SafeZone_Knockout_TA.EZoneEvent
enum class EZoneEvent : uint8_t
{
    ZE_None                      = 0,
    ZE_FinalZone                 = 1,
    ZE_END                       = 2
};

// Enum TAGame.GFxData_ProductFilter_TA.EProductFilterType
enum class EProductFilterType : uint8_t
{
    ProductFilterType_TradeIn     = 0,
    ProductFilterType_P2P          = 1,
    ProductFilterType_Garage       = 2,
    ProductFilterType_PremiumInventory = 3,
    ProductFilterType_END          = 4
};

// Enum TAGame.RPC_GetActivePlaylists_TA.EActivePlaylistType
enum class EActivePlaylistType : uint8_t
{
    ActivePlaylistType_Unused      = 0,
    ActivePlaylistType_Standard     = 1,
    ActivePlaylistType_Extra        = 2,
    ActivePlaylistType_Arcade       = 3,
    ActivePlaylistType_LTM          = 4,
    ActivePlaylistType_LargeParty   = 5,
    ActivePlaylistType_Dueling      = 6,
    ActivePlaylistType_END          = 7
};

// Enum TAGame.TourStatus_TA.ETourStatus
enum class ETourStatus : uint8_t
{
    TourStatus_None                = 0,
    TourStatus_Cancelled           = 1,
    TourStatus_Full                = 2,
    TourStatus_END                 = 3
};

// Enum TAGame._UserBugReportTypes_TA.EUserBugReportCategoryType
enum class EUserBugReportCategoryType : uint8_t
{
    UBRCT_UserInput                = 0,
    UBRCT_AutoReport               = 1,
    UBRCT_CustomerSupport          = 2,
}

```

```

UBRCT_Parent          = 3,
UBRCT_END            = 4
};

// Enum TAGame._AITypes_TA.EAITeam
enum class EAITeam : uint8_t
{
    AITeam_Self        = 0,
    AITeam_Enemy       = 1,
    AITeam_Any         = 2,
    AITeam_END         = 3
};

// Enum TAGame._AITypes_TA.EBTStatus
enum class EBTStatus : uint8_t
{
    BTStatus_Fail      = 0,
    BTStatus_Success   = 1,
    BTStatus_Running   = 2,
    BTStatus_END        = 3
};

// Enum TAGame._CrowdTypes_TA.ECrowdValueType
enum class ECrowdValueType : uint8_t
{
    CVT_Excitement     = 0,
    CVT_Tension         = 1,
    CVT_Anticipation   = 2,
    CVT_Upset           = 3,
    CVT_END             = 4
};

// Enum TAGame._FreeplayCommandsTypes_TA.EFreeplayCommand
enum class EFreeplayCommand : uint8_t
{
    FreeplayCommand_BallInFront     = 0,
    FreeplayCommand_BallOnCar       = 1,
    FreeplayCommand_RedirectPass   = 2,
    FreeplayCommand_PopBallUp      = 3,
    FreeplayCommand_DefendShot    = 4,
    FreeplayCommand_END             = 5
};

// Enum TAGame._TrainingTypes_TA.ETrainingRoundAttempt
enum class ETrainingRoundAttempt : uint8_t
{
    TrainingRoundAttempt_NotAttempted = 0,
    TrainingRoundAttempt_Attempted   = 1,
    TrainingRoundAttempt_Completed   = 2,
    TrainingRoundAttempt_END         = 3
};

// Enum TAGame._TrainingTypes_TA.ETrainingNavigationCommand
enum class ETrainingNavigationCommand : uint8_t

```

```

{
TrainingNavigationCommand_SelectShot      = 0,
TrainingNavigationCommand_PreviousShot    = 1,
TrainingNavigationCommand_NextShot       = 2,
TrainingNavigationCommand_ToggleShuffle   = 3,
TrainingNavigationCommand_END            = 4
};

// Enum TAGame._TrainingTypes_TA.ETrainingManipulationCommand
enum class ETrainingManipulationCommand : uint8_t
{
TrainingManipulationCommand_MirrorShot    = 0,
TrainingManipulationCommand_END           = 1
};

// Enum TAGame.AchievementManager_TA.EAchievementGame
enum class EAchievementGame : uint8_t
{
AG_Virtuoso                           = 0,
AG_Stocked                            = 1,
AG_FarFarAway                          = 2,
AG_SuperVictorious                    = 3,
AG_Champion                           = 4,
AG_TheStreak                          = 5,
AG_HelensPride                        = 6,
AG_BattleCarCollector                 = 7,
AG_DropsintheBucket                  = 8,
AG_Rocketeer                          = 9,
AG_GreaseMonkey                       = 10,
AG_PitchVeteran                      = 11,
AG_RidersBlock                        = 12,
AG_BreakShot                           = 13,
AG_Turbocharger                       = 14,
AG_DrillSergeant                     = 15,
AG_MinutetoWinit                      = 16,
AG_SpeedDemon                         = 17,
AG_PickMeUp                            = 18,
AG_WallCrawler                        = 19,
AG_TeamPlayer                          = 20,
AG_SARPBCForever                      = 21,
AG_FeatherinYourRecap                = 22,
AG_Winner                             = 23,
AG_CleanSheet                         = 24,
AG_TripleThreat                       = 25,
AG_DoubleUp                           = 26,
AG_SinglesClub                        = 27,
AG_PerfectStart                        = 28,
AG_StillAShowOff                     = 29,
AG_KnowTheDrill                       = 30,
AG_Traveler                           = 31,
AG_Tinkerer                           = 32,
AG_FirstTimer                         = 33,
AG_BarrasBravas                      = 34,
AG_Friendly                           = 35,
};

```

```

AG_END          = 36
};

// Enum TAGame.AchievementManager_TA.EAchievementEXP0
enum class EAchievementEXP0 : uint8_t
{
AEXP0_SkyHigh      = 0,
AEXP0_AllFours    = 1,
AEXP0_Gladiator     = 2,
AEXP0_END          = 3
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC0
enum class EAchievementDLC0 : uint8_t
{
ADLC0_WinningIsWinning    = 0,
ADLC0_AnInchAnd62Miles    = 1,
ADLC0_RideOrDie          = 2,
ADLC0_DontLookBack        = 3,
ADLC0_FamilyNotFriends    = 4,
ADLC0_DriftKing           = 5,
ADLC0_END                 = 6
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC1
enum class EAchievementDLC1 : uint8_t
{
ADLC1_SurvivalOfTheFittest = 0,
ADLC1_Heartbreaker         = 1,
ADLC1_NaturalProgression   = 2,
ADLC1_Throwback             = 3,
ADLC1_HotShotPartTwo        = 4,
ADLC1_END                  = 5
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC2
enum class EAchievementDLC2 : uint8_t
{
ADLC2_MyWorldIsFire        = 0,
ADLC2_Spectacular           = 1,
ADLC2_Savage                 = 2,
ADLC2_Ruthless                = 3,
ADLC2_PsychoMasterExploder    = 4,
ADLC2_END                  = 5
};

// Enum TAGame.AchievementManager_TA.EAchievementXB00
enum class EAchievementXB00 : uint8_t
{
AXB00_NiceHat              = 0,
AXB00_MarcusVMasterChief     = 1,
AXB00_END                  = 2
};

```

```
// Enum TAGame.AchievementManager_TA.EAchievementDLC3
enum class EAchievementDLC3 : uint8_t
{
ADLC3_MadScientist          = 0,
ADLC3_IcingTheCake          = 1,
ADLC3_LeftWingRightWing    = 2,
ADLC3_FastBreak              = 3,
ADLC3_BuzzerBeater          = 4,
ADLC3_BuddingArtist          = 5,
ADLC3_OneBetter              = 6,
ADLC3_Certifiable            = 7,
ADLC3_END                    = 8
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC4
enum class EAchievementDLC4 : uint8_t
{
ADLC4_GG                     = 0,
ADLC4_Trifecta                = 1,
ADLC4_InfinitePower           = 2,
ADLC4_StoppedCold              = 3,
ADLC4_SeaTurtle                = 4,
ADLC4_GetUpMrBubbles           = 5,
ADLC4_RocketGenocider          = 6,
ADLC4_END                      = 7
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC5
enum class EAchievementDLC5 : uint8_t
{
ADLC5_RegisteredVoter          = 0,
ADLC5_Metaverse                 = 1,
ADLC5_BraveTheElements          = 2,
ADLC5_DamageControl              = 3,
ADLC5_FullCourse                = 4,
ADLC5_BuckminsterX10           = 5,
ADLC5_END                      = 6
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC6
enum class EAchievementDLC6 : uint8_t
{
ADLC6_StormTrooper              = 0,
ADLC6_GoodTimes                  = 1,
ADLC6_SwapMeet      NewProfileWhoThis      = 2,
ADLC6_TradeSecret                = 3,
ADLC6_RankUp                    = 4,
ADLC6_ComingOnStrong             = 5,
ADLC6_END                      = 6
};

// Enum TAGame.AchievementManager_TA.EAchievementDLC7
enum class EAchievementDLC7 : uint8_t
{
```

```

ADLC7_JoinTheClub          = 0,
ADLC7_TogetherIsBetter    = 1,
ADLC7_NewChallenger       = 2,
ADLC7_PeoplePerson         = 3,
ADLC7_SquadGoals           = 4,
ADLC7_BestOfTheBunch      = 5,
ADLC7_END                  = 6
};

// Enum TAGame.AchievementManager_TA.EAchievementMeta
enum class EAchievementMeta : uint8_t
{
AMeta_PointCollector        = 0,
AMeta_END                   = 1
};

// Enum TAGame.AchievementSave_TA.EAchievementSaveVersion
enum class EAchievementSaveVersion : uint8_t
{
AchievementSaveVersion_FreeToPlayLaunch     = 0,
AchievementSaveVersion_END                 = 1
};

// Enum TAGame.ActivityFeedManager_TA.EActivityFeedType
enum class EActivityFeedType : uint8_t
{
AF_UnlockedBody              = 0,
AF_DrivenDistance            = 1,
AF_ScoredGoal                = 2,
AF_BlockedShot                = 3,
AF_WonGame                    = 4,
AF_MVPScore                  = 5,
AF_DefeatedFriend             = 6,
AF_CompletedRegularSeason     = 7,
AF_WonChampionship            = 8,
AF_UnlockedItem                = 9,
AF_UnlockedSweetTooth          = 10,
AF_END                         = 11
};

// Enum TAGame.AnimNotify_SetBoneVisibility_TA.VisibilityType
enum class EVisibilityType : uint8_t
{
VT_Socket                     = 0,
VT_Bone                       = 1,
VT_END                         = 2
};

// Enum TAGame.GameplayMusicPlayer_TA.EPlaylistType
enum class EPlaylistType : uint8_t
{
PlaylistType_Training          = 0,
PlaylistType_Match              = 1,
PlaylistType_END                = 2
};

```

```

};

// Enum TAGame.GameplayMusicPlayer_TA.EMusicPlayerState
enum class EMusicPlayerState : uint8_t
{
    MusicPlayerState_None          = 0,
    MusicPlayerState_Enabled       = 1,
    MusicPlayerState_Muted         = 2,
    MusicPlayerState_Disabled     = 3,
    MusicPlayerState_END           = 4
};

// Enum TAGame.AssetAttribute_ForceWheelAxele_TA.EAxeleToForce
enum class EAxeleToForce : uint8_t
{
    ForceFrontAxele              = 0,
    ForceBackAxele                = 1,
    EAxeleToForce_END             = 2
};

// Enum TAGame.BackFillPolicy_TA.EPolicyType
enum class EPolicyType : uint8_t
{
    PolicyType_Soccar             = 0,
    PolicyType_Knockout            = 1,
    PolicyType_END                 = 2
};

// Enum TAGame.PitchTekDrawingComponent_TA.EPitchTekDecalType
enum class EPitchTekDecalType : uint8_t
{
    PTDT_Ball                     = 0,
    PTDT_Car                      = 1,
    PTDT_END                      = 2
};

// Enum TAGame.BotConfig_TA.EBotSkill
enum class EBotSkill : uint8_t
{
    BotSkill_Intro                 = 0,
    BotSkill_Easy                  = 1,
    BotSkill_Medium                = 2,
    BotSkill_Hard                  = 3,
    BotSkill_Unfair                = 4,
    BotSkill_END                   = 5
};

// Enum TAGame.BreakOutActor_Platform_TA.EBreakoutDamageState
enum class EBreakoutDamageState : uint8_t
{
    DamageState_Start               = 0,
    DamageState_Damaged              = 1,
    DamageState_Broken                = 2,
    DamageState_END                  = 3
};

```

};

```
// Enum TAGame.BTLock.ELockScope
enum class ELockScope : uint8_t
{
    LockScope_Local           = 0,
    LockScope_Global          = 1,
    LockScope_END              = 2
};
```

```
// Enum TAGame.CameraState_CameraTrack_TA.ECameraTrackFlyType
enum class ECameraTrackFlyType : uint8_t
{
    CameraTrackFly_Never      = 0,
    CameraTrackFly_WhenMoving   = 1,
    CameraTrackFly_WhenPaused    = 2,
    CameraTrackFly_END          = 3
};
```

```
// Enum TAGame.CameraState_DirectorMoving_TA.ECameraLoopType
enum class ECameraLoopType : uint8_t
{
    CLT_StopAtEnd            = 0,
    CLT_PingPong              = 1,
    CLT_Loop                  = 2,
    CLT_END                   = 3
};
```

```
// Enum TAGame.CameraState_ReplayAutoCam_TA.ECountDownFocus
enum class ECountDownFocus : uint8_t
{
    CDF_None                 = 0,
    CDF_Pan                  = 1,
    CDF_Rotate                = 2,
    CDF_END                   = 3
};
```

```
// Enum TAGame.CameraStateSelector_Priority_TA.EStatePriority
enum class EStatePriority : uint8_t
{
    SP_Default                = 0,
    SP_OverrideAny             = 1,
    SP_OverrideLower            = 2,
    SP_InitialOnly              = 3,
    SP_END                     = 4
};
```

```
// Enum TAGame.ItemDropGroup_TA.EDropGroupDisplayOrder
enum class EDropGroupDisplayOrder : uint8_t
{
    DropGroupDisplayOrder_Default        = 0,
    DropGroupDisplayOrder_Challenge       = 1,
    DropGroupDisplayOrder_RewardDrop      = 2,
    DropGroupDisplayOrder_GarageSlot       = 3,
};
```

```

DropGroupDisplayOrder_LevelUpReward      = 4,
DropGroupDisplayOrder_Currency          = 5,
DropGroupDisplayOrder_Container         = 6,
DropGroupDisplayOrder_END              = 7
};

// Enum TAGame.ChallengeFolder_TA.EChallengeSort
enum class EChallengeSort : uint8_t
{
ECS_None                           = 0,
ECS_TopBottom                      = 1,
ECS_END                            = 2
};

// Enum TAGame.ClientConnectionTracker_TA.EConnectionQualityState
enum class EConnectionQualityState : uint8_t
{
CQS_Good                           = 0,
CQS_Mediocre                       = 1,
CQS_Bad                            = 2,
CQS_END                            = 3
};

// Enum TAGame.CrumbTrailRedefinition_TA.CrumbRedefinitionType
enum class ECrumbRedefinitionType : uint8_t
{
CR_TrailData                       = 0,
CR_Triggers                         = 1,
CR_CompletionTriggers               = 2,
CR_Crumbs                           = 3,
CR_END                             = 4
};

// Enum TAGame.CrumbTrailRedefinition_TA.CrumbAction
enum class ECrumbAction : uint8_t
{
CA_Add                             = 0,
CA_Remove                          = 1,
CA_Redo                            = 2,
CA_END                             = 3
};

// Enum TAGame.DistributionFloatSpeed_TA.ESpeedDistributionSpace
enum class ESspeedDistributionSpace : uint8_t
{
SDD_Forward                        = 0,
SDD_Side                           = 1,
SDD_Up                             = 2,
SDD_World                          = 3,
SDD_END                            = 4
};

// Enum TAGame.EngagementEventsConfig_TA.EEngagementEventType
enum class EEngagementEventType : uint8_t

```

```

{
EngagementEventType_DoubleXP          = 0,
EngagementEventType_DoublePaint       = 1,
EngagementEventType_END              = 2
};

// Enum TAGame.EOSVoiceSettingsSave_TA.EVoiceSettingsSaveVersion
enum class EVoiceSettingsSaveVersion : uint8_t
{
EVoiceSettingsSaveVersion_VoiceInputMode    = 0,
EVoiceSettingsSaveVersion_VoiceChatFilter   = 1,
EVoiceSettingsSaveVersion_END              = 2
};

// Enum TAGame.FirstTimeExperienceRedefinition_TA.FTEReDefinitionAction
enum class EFTEReDefinitionAction : uint8_t
{
RT_Rename                  = 0,
RT_Add                     = 1,
RT_Remove                  = 2,
RT_END                     = 3
};

// Enum TAGame.FirstTimeExperienceRedefinition_TA.FTEReDefinitionType
enum class EFTEReDefinitionType : uint8_t
{
ER_Group                   = 0,
ER_Checkpoint               = 1,
ER_Triggers                 = 2,
ER_UserGroup                = 3,
ER_SaveGroup                = 4,
ER_SaveAll                  = 5,
ER_END                      = 6
};

// Enum TAGame.ForceVolume_TA.EForceDirection
enum class EForceDirection : uint8_t
{
EFD_VolumeRotation          = 0,
EFD_ActorVelocity           = 1,
EFD_Custom                  = 2,
EFD_END                     = 3
};

// Enum TAGame.FXActor_Boost_TA.EmitterSocketBehavior
enum class EmitterSocketBehavior : uint8_t
{
ESB_Default                 = 0,
ESB_SingleEmitterAveragePosition = 1,
ESB_END                     = 2
};

// Enum TAGame.FXActor_Car_Knockout_TA.EStateEvent
enum class EStateEvent : uint8_t

```

```

{
SE_Block = 0,
SE_AttackStunned = 1,
SE_BlockStunned = 2,
SE_TradeStunned = 3,
SE_GrabStunned = 4,
SE_Grabbed = 5,
SE_Respawn = 6,
SE_END = 7
};

// Enum TAGame.FXActor_Knockout_Attack_TA.EAttackType
enum class EAttackType : uint8_t
{
AT_None = 0,
AT_Light = 1,
AT_Heavy = 2,
AT_END = 3
};

// Enum TAGame.GameEditor_Ring_TA.EDetectionType
enum class EDetectionType : uint8_t
{
DT_Ball = 0,
DT_Player = 1,
DT_END = 2
};

// Enum TAGame.GameEvent_Tutorial_TA.ERotationType
enum class ERotationType : uint8_t
{
Rot_Auto = 0,
Rot_AutoAim = 1,
Rot_AutoForward = 2,
Rot_AutoArch = 3,
Rot_World = 4,
Rot_MapCenter = 5,
Rot_END = 6
};

// Enum TAGame.GameEvent_TrainingEditor_TA.EPlayTestType
enum class EPlayTestType : uint8_t
{
PlayTest_LoopRound = 0,
PlayTest_IncrementRound = 1,
PlayTest_IncrementRoundLoop = 2,
PlayTest_END = 3
};

// Enum TAGame.GameplaySettingsSave_TA.EGameplaySettingsSaveVersion
enum class EGameplaySettingsSaveVersion : uint8_t
{
EGameplaySettingsSaveVersion_ChatFilterDisambiguation = 0,
EGameplaySettingsSaveVersion_END = 1
};

```

```

};

// Enum TAGame.GFxData_AdHoc_TA.EAdHocState
enum class EAdHocState : uint8_t
{
AHS_Enabled           = 0,
AHS_Disabled          = 1,
AHS_END               = 2
};

// Enum TAGame.GFxData_CarKnockOut_TA.EStunlockType
enum class EStunlockType : uint8_t
{
SL_None               = 0,
SL_Stunned            = 1,
SL_Grabbed             = 2,
SL_END                = 3
};

// Enum TAGame.GFxData_ChallengeManager_TA.EChallengeFilter
enum class EChallengeFilter : uint8_t
{
CF_Default            = 0,
CF_Progress           = 1,
CF_END                = 2
};

// Enum TAGame.GFxData_EngagementEventsConfig_TA.ENotificationType
enum class ENotificationType : uint8_t
{
NotificationType_Start      = 0,
NotificationType_End         = 1,
NotificationType_END         = 2
};

// Enum TAGame.GFxData_Friends_TA.ENameDisplayRelationshipProxy
enum class ENameDisplayRelationshipProxy : uint8_t
{
NameDisplayRelationshipProxy_NoChange    = 0,
NameDisplayRelationshipProxy_ForcePlatform = 1,
NameDisplayRelationshipProxy_ForcePsyNet   = 2,
NameDisplayRelationshipProxy_ForceAll     = 3,
NameDisplayRelationshipProxy_END         = 4
};

// Enum TAGame.GFxData_LegacyStatus_TA.ELegacyStatusSetState
enum class ELegacyStatusSetState : uint8_t
{
LSSS_NotSet            = 0,
LSSS_Updating           = 1,
LSSS_UnableToUpdate     = 2,
LSSS_Set                = 3,
LSSS_END                = 4
};

```

```

// Enum TAGame.GFxData_LocalPlayer_TA.ESplitScreenPosition
enum class ESplitScreenPosition : uint8_t
{
    SSP_Center          = 0,
    SSP_Left            = 1,
    SSP_Right           = 2,
    SSP_Top             = 3,
    SSP_Bottom          = 4,
    SSP_TopLeft         = 5,
    SSP_TopRight        = 6,
    SSP_BottomLeft      = 7,
    SSP_BottomRight     = 8,
    SSP_END             = 9
};

// Enum TAGame.GFxData_MapPrefs_TA.EMapPrefType
enum class EMapPrefType : uint8_t
{
    MapPrefType_None    = 0,
    MapPrefType_Like    = 1,
    MapPrefType_Dislike = 2,
    MapPrefType_END     = 3
};

// Enum TAGame.GFxData_MenuTreeNode_TA.ENodeBannerType
enum class ENodeBannerType : uint8_t
{
    NodeBannerType_None    = 0,
    NodeBannerType_Unavailable = 1,
    NodeBannerType_Searching = 2,
    NodeBannerType_Locked   = 3,
    NodeBannerType_END     = 4
};

// Enum TAGame.GFxData_OnlineMatchStatus_TA.SearchMessageType
enum class ESearchMessageType : uint8_t
{
    MessageType_Status    = 0,
    MessageType_Warning    = 1,
    MessageType_Error      = 2,
    MessageType_Countdown  = 3,
    MessageType_END        = 4
};

// Enum TAGame.GFxData_Party_TA.EPsyNetPartyStatus
enum class EPsyNetPartyStatus : uint8_t
{
    PsyNetParty_Disabled   = 0,
    PsyNetParty_Connecting  = 1,
    PsyNetParty_Connected   = 2,
    PsyNetParty_Disconnected = 3,
    PsyNetParty_END         = 4
};

```

```

// Enum TAGame.GFxData_ProductFilter_TA.EFilterProductAttribute
enum class EFilterProductAttribute : uint8_t
{
    FPA_Indestructible          = 0,
    FPA_END                      = 1
};

// Enum TAGame.GFxData_QuickChatBindings_TA.EQuickChatState
enum class EQuickChatState : uint8_t
{
    QCS_InGame                  = 0,
    QCS_PostGame                 = 1,
    QCS_PreGame                  = 2,
    QCS_END                      = 3
};

// Enum TAGame.GFxData_Restrictions_TA.EReasonType
enum class EReasonType : uint8_t
{
    RT_NotRestricted            = 0,
    RT_Credits                  = 1,
    RT_ESportsTokens            = 2,
    RT_Merchandise               = 3,
    RT_END                      = 4
};

// Enum TAGame.GFxData_UserSetting_TA.EUserSettingType
enum class EUserSettingType : uint8_t
{
    UserSettingType_List          = 0,
    UserSettingType_Boolean        = 1,
    UserSettingType_Scalar         = 2,
    UserSettingType_END            = 3
};

// Enum TAGame.ShopTabs_TA.EShopTabType
enum class EShopTabType : uint8_t
{
    ShopTabType_ShopCatalogueTable      = 0,
    ShopTabType_EMtxCatalogCategory     = 1,
    ShopTabType_Esports                = 2,
    ShopTabType_END                    = 3
};

// Enum TAGame.SpecialEventConfig_TA.ESpecialEventState
enum class ESpecialEventState : uint8_t
{
    SES_Inactive                  = 0,
    SES_Active                     = 1,
    SES_Redemption                 = 2,
    SES_END                        = 3
};

```

```
// Enum TAGame.GFxData_TourEvent_TA.ETourEventStatus
enum class ETourEventStatus : uint8_t
{
    TES_NonePending = 0,
    TES_Registered = 1,
    TES_CheckInOpen = 2,
    TES_CheckedIn = 3,
    TES_GeneratingBrackets = 4,
    TES_MatchPending = 5,
    TES_MatchReady = 6,
    TES_InMatch = 7,
    TES_MatchResultsPending = 8,
    TES_Paused = 9,
    TES_MatchEnding = 10,
    TES_TroubleJoining = 11,
    TES_END = 12
};
```

```
// Enum TAGame.GFxData_TrainingModeBrowser_TA.ETrainingModeSearchTypes
enum class ETrainingModeSearchTypes : uint8_t
{
    TMST_AllPublic = 0,
    TMST_FeaturedOnly = 1,
    TMST_END = 2
};
```

```
// Enum TAGame.GFxData_UserBugReport_TA.EUserBugReportSubmitStatus
enum class EUserBugReportSubmitStatus : uint8_t
{
    UBRSS_None = 0,
    UBRSS_Generated = 1,
    UBRSS_Submitting = 2,
    UBRSS_Success = 3,
    UBRSS_Error = 4,
    UBRSS_END = 5
};
```

```
// Enum TAGame.GFxEngine_TA.ESoundStatePriority
enum class ESoundStatePriority : uint8_t
{
    SSP_None = 0,
    SSP_Scoreboard = 1,
    SSP_Replay = 2,
    SSP_MidGameMenu = 3,
    SSP_Pause = 4,
    SSP_BeginRound = 5,
    SSP_EndRound = 6,
    SSP_WinnerMenu = 7,
    SSP_MainMenu = 8,
    SSP_PremiumGarage = 9,
    SSP_Roulette = 10,
    SSP_AudioPreview = 11,
    SSP_MusicPreview = 12,
    SSP_ThirdPartyOverlay = 13
};
```

```

SSP>LoadingScreen          = 14,
SSP>END                  = 15
};

// Enum TAGame.StatGraph_TA.EStatGraphLayer
enum class EStatGraphLayer : uint8_t
{
StatGraphLayer_Background      = 0,
StatGraphLayer_Text            = 1,
StatGraphLayer_Graph           = 2,
StatGraphLayer_END             = 3
};

// Enum TAGame.InterpTrackSkelControlRotation_TA.ERotationAxis
enum class ERotationAxis : uint8_t
{
ROTAXIS_Yaw                  = 0,
ROTAXIS_Pitch                 = 1,
ROTAXIS_Roll                  = 2,
ROTAXIS_END                   = 3
};

// Enum TAGame.MenuSequencer_TA.ESequenceStatus
enum class ESequenceStatus : uint8_t
{
SS_Idle                      = 0,
SS_Entering                  = 1,
SS_Leaving                    = 2,
SS_Pending                    = 3,
SS_END                        = 4
};

// Enum TAGame.MenuTreeNode_TA.ENodeEnabledRequirement
enum class ENodeEnabledRequirement : uint8_t
{
NodeEnabledRequirement_InMainMenu    = 0,
NodeEnabledRequirement_RankedEnabled = 1,
NodeEnabledRequirement_END           = 2
};

// Enum TAGame.MenuTreeNode_TA.ENodeBadgeType
enum class ENodeBadgeType : uint8_t
{
NodeBadgeType_None              = 0,
NodeBadgeType_Default            = 1,
NodeBadgeType_Success            = 2,
NodeBadgeType_END                = 3
};

// Enum TAGame.MenuTreeNode_Playlist_TA.EPlaylistSpecialType
enum class EPlaylistSpecialType : uint8_t
{
PlaylistSpecialType_None        = 0,
PlaylistSpecialType_Dueling     = 1,

```

```

PlaylistSpecialType_LTM          = 2,
PlaylistSpecialType_END         = 3
};

// Enum TAGame.MusicPlayerSave_TA.EMusicPlayerSave
enum class EMusicPlayerSave : uint8_t
{
MusicPlayerSaveVersion_NewGameplayMusicSettings = 0,
MusicPlayerSaveVersion_END           = 1
};

// Enum TAGame.MusicPlayerSave_TA.EGameplayMusicSetting
enum class EGameplayMusicSetting : uint8_t
{
GameplayMusic_Off                = 0,
GameplayMusic_TraningOnly        = 1,
GameplayMusic_MatchesOnly        = 2,
GameplayMusic_AlwaysOn           = 3,
GameplayMusic_Undefined          = 4,
GameplayMusic_END                = 5
};

// Enum TAGame.OrbitHitHandler_TA.EOrbitCenterLocation
enum class EOrbitCenterLocation : uint8_t
{
OCB_BallHitLocation            = 0,
OCB_CenterOfGoal               = 1,
OCB_END                         = 2
};

// Enum TAGame.OrbitHitHandler_TA.EOrbitAxisBehavior
enum class EOrbitAxisBehavior : uint8_t
{
OAB_StaticRotationAxis          = 0,
OAB_RandomRotationAxis          = 1,
OAB_END                         = 2
};

// Enum TAGame.OutOfWorldVolume_TA.EOutOfWorld
enum class EOutOfWorld : uint8_t
{
OOW_Touch                        = 0,
OOW_Untouch                      = 1,
OOW_END                          = 2
};

// Enum TAGame.OverrideMaterialsHitHandler_TA.OverrideMaterialMode_TA
enum class EOverrideMaterialMode_TA : uint8_t
{
OverrideMaterial_SkinOnly         = 0,
OverrideMaterial_ChassisOnly      = 1,
OverrideMaterial_Everything       = 2,
OverrideMaterial_END              = 3
};

```

```

// Enum TAGame.ProductAssetLoader_TA.EAssetLoadState
enum class EAssetLoadState : uint8_t
{
    AssetLoadState_New = 0,
    AssetLoadState_WaitingAsyncLoad = 1,
    AssetLoadState_AsyncLoad = 2,
    AssetLoadState_Loaded = 3,
    AssetLoadState_END = 4
};

// Enum TAGame.ProductAttribute_Trademark_TA.ETrademarkGlyph
enum class ETrademarkGlyph : uint8_t
{
    TrademarkGlyph_TradeMark = 0,
    TrademarkGlyph_Registered = 1,
    TrademarkGlyph_Copyright = 2,
    TrademarkGlyph_END = 3
};

// Enum TAGame.ProductEquipProfileSlot_Custom_TA.EEnableSlotCustomization
enum class EEnableSlotCustomization : uint8_t
{
    EnableSlotCustomization_Enabled = 0,
    EnableSlotCustomization_Disabled = 1,
    EnableSlotCustomization_END = 2
};

// Enum TAGame.ProductsArchiveSave_TA.EProductsArchiveSaveVersion
enum class EProductsArchiveSaveVersion : uint8_t
{
    ProductsArchiveSaveVersion_ConvertHashIDsToInstanceIDs = 0,
    ProductsArchiveSaveVersion_InstanceIDV2 = 1,
    ProductsArchiveSaveVersion_END = 2
};

// Enum TAGame.ProductsFavoriteSave_TA.EProductsFavoriteSaveVersion
enum class EProductsFavoriteSaveVersion : uint8_t
{
    EProductsFavoriteSaveVersion_ConvertHashIDsToInstanceIDs = 0,
    EProductsFavoriteSaveVersion_InstanceIDV2 = 1,
    EProductsFavoriteSaveVersion_END = 2
};

// Enum TAGame.ProductsSave_TA.EProductsSaveVersion
enum class EProductsSaveVersion : uint8_t
{
    EProductsSaveVersion_InstanceIDV2 = 0,
    EProductsSaveVersion_END = 1
};

// Enum TAGame.ProfileControlsSave_TA.EProfilePCSaveVersion
enum class EProfileControlsSave_TA_EProfilePCSaveVersion : uint8_t
{
}

```

```

ProfileControlsSaveVersion_NewVibrationSettings = 0,
ProfileControlsSaveVersion_END = 1
};

// Enum TAGame.ProfileGamepadSave_TA.EProfileGamepadSaveVersion
enum class EProfileGamepadSaveVersion : uint8_t
{
ProfileGamepadSaveVersion_NewDefaultControls = 0,
ProfileGamepadSaveVersion_END = 1
};

// Enum TAGame.ProfileGameplaySave_TA.EProfileGameplaySaveVersion
enum class EProfileGameplaySaveVersion : uint8_t
{
ProfileGameplaySaveVersion_RankInfoDisplayType = 0,
ProfileGameplaySaveVersion_END = 1
};

// Enum TAGame.ProfileLoadoutSave_TA.ELoadoutSaveVersion
enum class ELoadoutSaveVersion : uint8_t
{
ELoadoutSaveVersion_InstanceIDV2 = 0,
ELoadoutSaveVersion_END = 1
};

// Enum TAGame.ProfilePCSave_TA.EProfilePCSaveVersion
enum class EProfilePCSave_TA_EProfilePCSaveVersion : uint8_t
{
ProfilePCSaveVersion_NewDefaultControls = 0,
ProfilePCSaveVersion_END = 1
};

// Enum TAGame.ProfileQuickChatSave_TA.EQuickChatVersion
enum class EQuickChatVersion : uint8_t
{
QuickChatVersion_AddPreGameQuickChat = 0,
QuickChatVersion_END = 1
};

// Enum TAGame.Replay_TA.EReplayVersion
enum class EReplayVersion : uint8_t
{
ReplayVersion_Base = 0,
ReplayVersion_OnlineLoadout = 1,
ReplayVersion_CameraSettings = 2,
ReplayVersion_TeamLoadouts = 3,
ReplayVersion_MatchType = 4,
ReplayVersion_ActorName = 5,
ReplayVersion_MutatorSeekFree = 6,
ReplayVersion_ClubColors = 7,
ReplayVersion_CameraTrack = 8,
ReplayVersion-END = 9
};

```

```
// Enum TAGame.Replay_TA.EReplayState
enum class EReplayState : uint8_t
{
    ReplayState_Idle          = 0,
    ReplayState_Recording      = 1,
    ReplayState_Playing        = 2,
    ReplayState_END            = 3
};

// Enum TAGame.SampleHistory_TA.EGraphSummaryType
enum class EGraphSummaryType : uint8_t
{
    GraphSummaryType_Latest    = 0,
    GraphSummaryType_Avg       = 1,
    GraphSummaryType_MinValue  = 2,
    GraphSummaryType_MaxValue  = 3,
    GraphSummaryType_Sum       = 4,
    GraphSummaryType_END       = 5
};

// Enum TAGame.SeasonSave_TA.ESeasonSaveVersion
enum class ESeasonSaveVersion : uint8_t
{
    SeasonSaveVersion_AddIntroBotDifficulty = 0,
    SeasonSaveVersion_END                  = 1
};

// Enum TAGame.SeqAct_CheckRotationAngle_TA.ECheckAxis
enum class ECheckAxis : uint8_t
{
    ECA_Pitch          = 0,
    ECA_Yaw            = 1,
    ECA_Roll           = 2,
    ECA_END            = 3
};

// Enum TAGame.SeqAct_SetLoadout_TA.EForcedTeamOverride
enum class EForcedTeamOverride : uint8_t
{
    ForcedTeamOverride_None   = 0,
    ForcedTeamOverride_Blue   = 1,
    ForcedTeamOverride_Orange = 2,
    ForcedTeamOverride_END    = 3
};

// Enum TAGame.SkelControlSingleBoneCopy_TA.ECopyAxis
enum class ECopyAxis : uint8_t
{
    CopyAxis_X          = 0,
    CopyAxis_Y          = 1,
    CopyAxis_Z          = 2,
    CopyAxis_Pitch       = 3,
    CopyAxis_Yaw         = 4,
    CopyAxis_Roll        = 5
};
```

```

CopyAxis-END = 6
};

// Enum TAGame.SpecialPickup_Spring_TA.ESpringState
enum class ESpringState : uint8_t
{
    SpringState_Traveling = 0,
    SpringState_PostHit = 1,
    SpringState_END = 2
};

// Enum TAGame.SpecialPickup_BallFreeze_TA.BallFreezeState
enum class EBallFreezeState : uint8_t
{
    BallFreezeState_Freezing = 0,
    BallFreezeState_Frozen = 1,
    BallFreezeState_END = 2
};

// Enum TAGame.SpecialPickup_Football_TA.EThrowSetting
enum class EThrowSetting : uint8_t
{
    ETS_Forward = 0,
    ETS_Backward = 1,
    ETS_Side = 2,
    ETS_END = 3
};

// Enum TAGame.SpecialPickup_GrapplingHook_TA.ERopeState
enum class ERopeState : uint8_t
{
    RopeState_Traveling = 0,
    RopeState_Attached = 1,
    RopeState_END = 2
};

// Enum TAGame.StatFactory_TA.EFiveHitType
enum class EFiveHitType : uint8_t
{
    FiveHit_None = 0,
    FiveHit_Low = 1,
    FiveHit_High = 2,
    FiveHit_END = 3
};

// Enum TAGame.StatTitle_TA.EStatTrackingUnitType
enum class EStatTrackingUnitType : uint8_t
{
    CountUnit = 0,
    SpeedUnit = 1,
    DistanceUnit = 2,
    EStatTrackingUnitType_END = 3
};

```

```

// Enum TAGame.StatusTrigger_Numerical_TA.TriggerConditionNumericalType
enum class ETriggerConditionNumericalType : uint8_t
{
    TCNT_GreaterThan          = 0,
    TCNT_LessThan              = 1,
    TCNT_Equal                 = 2,
    TCNT_GreaterEqual          = 3,
    TCNT_LessThanEqual         = 4,
    TCNT_END                   = 5
};

// Enum TAGame.Tutorial_TA.ETutorialOutput
enum class ETutorialOutput : uint8_t
{
    TutorialOutput_Ended        = 0,
    TutorialOutput_Completed     = 1,
    TutorialOutput_Failed        = 2,
    TutorialOutput_ScreenFadedIn   = 3,
    TutorialOutput_ScreenFadedOut    = 4,
    TutorialOutput_AllMessagesDisplayed = 5,
    TutorialOutput_Started        = 6,
    TutorialOutput_Reset          = 7,
    TutorialOutput_Skipped         = 8,
    TutorialOutput_AnswerRight     = 9,
    TutorialOutput_AnswerWrong      = 10,
    TutorialOutput_END            = 11
};

// Enum TAGame.Tutorial_TA.ETutorialStatus
enum class ETutorialStatus : uint8_t
{
    TutorialStatus_None          = 0,
    TutorialStatus_Failed         = 1,
    TutorialStatus_Completed       = 2,
    TutorialStatus_END            = 3
};

// Enum TAGame.Tutorial_TA.EPlayerDriveType
enum class EPlayerDriveType : uint8_t
{
    PlayerDriveType_All           = 0,
    PlayerDriveType_Player         = 1,
    PlayerDriveType_Bot            = 2,
    PlayerDriveType_END            = 3
};

// Enum TAGame.VehiclePickup_Boost_TA.EBoostType
enum class EBoostType : uint8_t
{
    BoostType_Unknown             = 0,
    BoostType_Pill                 = 1,
    BoostType_Pad                  = 2,
    BoostType_END                  = 3
};

```

```
/*
#
=====
===== #
# Classes
#
=====
===== #
*/
// Class TAGame._AITypes_TA
// 0x0000 (0x0060 - 0x0060)
class U_AITypes_TA : public UObject
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._AITypes_TA");
}

return uClassPointer;
};

// Class TAGame._ShopTypes_TA
// 0x0000 (0x0060 - 0x0060)
class U_ShopTypes_TA : public UObject
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._ShopTypes_TA");
}

return uClassPointer;
};

};
```

```

// Class TAGame._Types_TA
// 0x0048 (0x0060 - 0x00A8)
class U_Types_TA : public UObject
{
public:
    struct FScriptDelegate           __AssetLoadedDelegate__Delegate;      // 0x0060
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __ProductThumbnailRenderedDelegate__Delegate; // 0x0078 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __MapImageLoadedDelegate__Delegate;      // 0x0090
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame._Types_TA");
        }

        return uClassPointer;
    }

    static class FString GetMessageForAutoUpdateReason(uint8_t Reason);
    static class FString GetKismetSafeString(class FString InValue);
    static int32_t GetTotalStatValue(struct FStatValue Stat);
    static uint8_t GetChatUsage(uint8_t Channel);
    static struct FLinearColor GammaCorrectColor(struct FLinearColor InColor, float Gamma);
    static struct FLoadoutTeamPaint GetLoadoutPaint(struct FLoadoutData InLoadoutData);
    static int32_t GetRandomColorID(class UCarColorSet_TA* Set);
    static void InitColorSet(class UCarColorSet_TA* Set, TArray<struct FGfxTeamColor>& OutColors);
    static class FString ProductQualityToString(uint8_t Quality);
    static bool ProductAttributesAreDifferent(struct FClientLoadoutOnlineData& A, struct FClientLoadoutOnlineData& B);
    static bool TeamPaintsAreDifferent(struct FLoadoutData& A, struct FLoadoutData& B);
    static bool OnlineProductsAreDifferent(TArray<struct FProductInstanceId>& A, TArray<struct FProductInstanceId>& B);
    static bool ProductsAreDifferent(TArray<int32_t>& A, TArray<int32_t>& B);
    static bool LoadoutsAreDifferent(struct FLoadoutData& A, struct FLoadoutData& B);
    static float UpdateInputValue(struct FInputRate Rate, float DesiredValue, float OldValue, float DeltaTime);
    static struct FClientLoadoutData ConvertToClientLoadout(struct FLoadoutData& FromData);
    void MapImageLoadedDelegate(struct FMapImageLoadResult Result);
    void ProductThumbnailRenderedDelegate(struct FProductThumbnailResult Result);
    void AssetLoadedDelegate(struct FAssetLoadResult Result);
    static struct FRandomWeight PickWeightedRandom(TArray<struct FRandomWeight> Arr, int32_t& Index);
    static bool VehicleInputHasChangedForKick(struct FVehicleInputs& NewInput, struct FVehicleInputs& OldInput);
};


```

```

// Class TAGame.AIController_TA
// 0x0268 (0x04A0 - 0x0708)
class AAIController_TA : public AAIController
{
public:
    class ACar_TA*           Car;          // 0x04A0 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    class AVehicle_TA*       Vehicle;     // 0x04A8 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    class UBTComponent*      BT;          // 0x04B0 (0x0008)
    [0x0000004004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UBTProxyType*      SelfProxyType; // 0x04B8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      NoneProxyType; // 0x04C0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      DebugProxyType; // 0x04C8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      TeamCarsProxyType; // 0x04D0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      EnemyCarsProxyType; // 0x04D8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      PlayerControlledCarsProxyType; // 0x04E0
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      OtherCarsProxyType; // 0x04E8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      BoostPillsProxyType; // 0x04F0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTProxyType*      SelectedProxyType; // 0x04F8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTarget*         SelfTarget;   // 0x0500 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTarget*         NoneTarget;  // 0x0508 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTarget*         IteratorTarget; // 0x0510 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTarget*         TeamCarsTarget; // 0x0518 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTarget*         SelectedTarget; // 0x0520 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTarget*         ContextTarget; // 0x0528 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class UBTNode*           DebugBTNode; // 0x0530 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTLockCollection* Locks;        // 0x0538 (0x0008)
    [0x0000004004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    struct FSkillRange       SkillSteerError; // 0x0540 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FSkillRange       SkillSteerSpeed; // 0x054C (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FSkillRange       SkillThrottleSpeed; // 0x0558 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    class AGameEvent_TA*     GameEvent;    // 0x0568 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    class AAIManager_TA*     AIManager;   // 0x0570 (0x0008)

```

```

[0x0000004000002000] (CPF_Transient)
struct FVehicleInputs Input; // 0x0578 (0x0020)
[0x0000000000002000] (CPF_Transient)
struct FVector ForwardDir; // 0x0598 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FRotator ForwardRotation; // 0x05A4 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FVector GroundNormal; // 0x05B0 (0x000C)
[0x0000000000002000] (CPF_Transient)
TArray<struct FCachedProxyData> CachedProxyDatas; // 0x05C0
(0x0010) [0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCachedTargetData> CachedTargetDatas; // 0x05D0
(0x0010) [0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FMap_Mirror TargetFilterMap; // 0x05E0 (0x0050)
[0x0000008000003000] (CPF_Native | CPF_Transient)
TArray<class UBTNode*> CachedTargetFilters; // 0x0630 (0x0010)
[0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FVector DebugDriveDestination; // 0x0640 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FCachedObstaclesData CachedObstacles; // 0x0650
(0x0028) [0x0000000000003000] (CPF_Native | CPF_Transient)
TArray<class ACar_TA*> TeamCars; // 0x0678 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class ACar_TA*> EnemyCars; // 0x0688 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class ACar_TA*> PlayerControlledCars; // 0x0698 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class ACar_TA*> OtherCars; // 0x06A8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long bForceUpdateConditions : 1; // 0x06B8 (0x0004)
[0x0000000000002002] [0x00000001] (CPF_Const | CPF_Transient)
unsigned long bCanUseBoost : 1; // 0x06B8 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long bBeingReplaced : 1; // 0x06B8 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
class UGameShare_TA* GameShare; // 0x06C0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class AActor* SelectedActor; // 0x06C8 (0x0008)
[0x0000000000002002] (CPF_Const | CPF_Transient)
float FieldScalar; // 0x06D0 (0x0004)
[0x0000000000002000] (CPF_Transient)
uint8_t ForcedTeam; // 0x06D4 (0x0001)
[0x0000000000000000]
struct FScriptDelegate __EventNamedEvent__Delegate; // 0x06D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSkillChanged__Delegate; // 0x06F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.AIController_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
void eventActivatePickup();
void SetSelectedActor(class AActor* NewActor);
void AllTargets(class UBTTTarget* Target, struct FAIProxyData& OutData);
bool GetObstacleInRadius(struct FVector Destination, float Radius, struct FVector& HitLocation);
bool GetObstacle(struct FVector Destination, struct FVector& HitLocation);
bool eventGetProxyDataInternal(class UBTProxyType* Type, int32_t Index, struct FAIProxyData& OutData);
bool GetIndexedProxyData(class UBTProxyType* Type, int32_t Index, struct FAIProxyData& OutData);
bool GetProxyData(class UBTProxyType* Type, struct FAIProxyData& OutData);
bool GetIndexedTargetData(class UBTTTarget* Target, int32_t Index, struct FAIProxyData& OutData);
bool GetTargetData(class UBTTTarget* Target, struct FAIProxyData& OutData);
struct FAIProxyData eventCreateVehicleProxyData(class AVehicle_TA* InVehicle);
struct FAIProxyData eventCreateProxyData(class AActor* A, float Radius);
float GetDriveTime(float Distance);
float GetDriveDistance(struct FAIProxyData Proxy);
struct FVector FlattenOffset(struct FVector Offset, float& Height);
struct FVector FlattenDestination(struct FVector Destination);
struct FAIProxyData FlattenPhysics(struct FAIProxyData& ProxyData);
float GetClosestTime2D(float MaxTime, struct FAIProxyData& A, struct FAIProxyData& B);
bool WillImpact2D(struct FAIProxyData& A, struct FAIProxyData& B, float& CollideTime);
void AdvancePhysics(float Time, struct FAIProxyData& PhysicsData);
void PredictPhysics(float Time, float GroundZ, struct FAIProxyData& PhysicsData);
void PredictLanding(float GroundZ, struct FAIProxyData& Data, float& Time);
struct FVector AdvanceLocation(float Time, struct FVector& Loc, struct FVector& Vel, struct FVector& Accel);
void DebugDriveTo(struct FVector Destination);
void DoNothing();
void AbortBT();
void eventDestroyed();
void OnCarSetup(class ACar_TA* InCar);
void SetCar();
void UnPossess();
void Possess(class APawn* inPawn);
void SetSkill(float NewSkill, float CanUseBoostThreshold);
void RelnitBT();
void HandleOtherCarDestroyed(class APawn_X* InCar);
void HandleOtherCarDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void HandleOtherCarTeamChanged(class ACar_TA* InCar);
void HandleCarSpawned(class AGameEvent_TA* InGameEvent, class ACar_TA* InCar);
void Init(class AGameEvent_TA* InGameEvent);
void eventPreBeginPlay();
void EventSkillChanged(class AAIController_TA* AI);
void EventNamedEvent(class AAIController_TA* AI, struct FName EventName, class AActor* Sender);

```

```

};

// Class TAGame.AIController_Soccar_TA
// 0x00B8 (0x0708 - 0x07C0)
class AAIController_Soccar_TA : public AAIController_TA
{
public:
    class UBTProxyType* [0x0000000000000001] (CPF_Edit) BallProxyType; // 0x0708 (0x0008)
    class UBTProxyType* [0x0000000000000001] (CPF_Edit) BallLandingProxyType; // 0x0710 (0x0008)
    class UBTProxyType* (0x0008) [0x0000000000000001] (CPF_Edit) BallTeamGoalLineProxyType; // 0x0718
    class UBTProxyType* (0x0008) [0x0000000000000001] (CPF_Edit) BallEnemyGoalLineProxyType; // 0x0720
    class UBTProxyType* [0x0000000000000001] (CPF_Edit) BallProjectedProxyType; // 0x0728 (0x0008)
    class UBTProxyType* [0x0000000000000001] (CPF_Edit) TeamGoalProxyType; // 0x0730 (0x0008)
    class UBTProxyType* [0x0000000000000001] (CPF_Edit) EnemyGoalProxyType; // 0x0738 (0x0008)
    class UBTProxyType* [0x0000000000000001] (CPF_Edit) MidFieldProxyType; // 0x0740 (0x0008)
    class UBTTTarget* [0x0000000000000001] (CPF_Edit) BallTarget; // 0x0748 (0x0008)
    class UBTTTarget* [0x0000000000000001] (CPF_Edit) TeamGoalTarget; // 0x0750 (0x0008)
    class UBTTTarget* [0x0000000000000001] (CPF_Edit) EnemyGoalTarget; // 0x0758 (0x0008)
    class AGameEvent_Soccar_TA* (0x0008) [0x000004000002000] (CPF_Transient) SoccarGame; // 0x0760
    class ABall_TA* [0x000004000002000] (CPF_Transient) Ball; // 0x0768 (0x0008)
    struct FAICachedGoalInfo [0x000004000082000] (CPF_Transient | CPF_Component) TeamGoalInfo; // 0x0770 (0x0018)
    struct FAICachedGoalInfo [0x000004000082000] (CPF_Transient | CPF_Component) EnemyGoalInfo; // 0x0788 (0x0018)
    float [0x00000000000002000] (CPF_Transient) LastBallHitTime; // 0x07A0 (0x0004)
    struct FScriptDelegate (0x0018) [0x00000000000400000] (CPF_NeedCtorLink) __EventBallChanged__Delegate; // 0x07A8

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AIController_Soccar_TA");
        }

        return uClassPointer;
    };
}

```

```

void DisablePickupAutoActivate();
bool eventGetProxyDataInternal(class UBTProxyType* Type, int32_t Index, struct FAIProxyData& OutData);
void GetGoalProxy(class UBTProxyType* Type, struct FAICachedGoalInfo GoalInfo, struct FAIProxyData& OutData);
void GetBallProxy(struct FAIProxyData& OutData);
void SetBall(class ABall_TA* InBall);
void HandleNewPickup();
void HandleHitBall(class ACar_TA* InCar, class ABall_TA* HitBall, struct FVector HitLocation,
struct FVector HitNormal);
void OnCarSetup(class ACar_TA* InCar);
void UpdateBall();
void HandleActiveRoundChanged(class AGameEvent_Soccar_TA* G);
void HandleBallsChanged(class AGameEvent_Soccar_TA* G, class ABall_TA* B);
struct FAICachedGoalInfo GetCachedGoalInfo(int32_t TeamNum);
void Init(class AGameEvent_TA* InGameEvent);
void EventBallChanged(class AAIController_Soccar_TA* SoccarAI);
};

// Class TAGame.AIManager_TA
// 0x0040 (0x0268 - 0x02A8)
class AAIController_TA : public AActor
{
public:
unsigned long          bFlushAILogEachLine : 1;           // 0x0268 (0x0004)
[0x0000000000004003] [0x00000001] (CPF_Edit | CPF_Const | CPF_Config)
unsigned long          bOutputLogToWindow : 1;           // 0x0268 (0x0004)
[0x0000000000004003] [0x00000002] (CPF_Edit | CPF_Const | CPF_Config)
unsigned long          bPauseForDirtyTrees : 1;           // 0x0268 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
class UBTLockCollection*      Locks;                  // 0x0270 (0x0008)
[0x00000000408000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_Component |
CPF_EditInline)
class AGameEvent_TA*        GameEvent;               // 0x0278 (0x0008)
[0x0000004000002000] (CPF_Transient)
TArray<class AAIController_TA*>    Bots;                  // 0x0280 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCachedNavMeshLocation>    CachedNavMeshLocations;      //
0x0290 (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class AFileLog*            AILogFile;                // 0x02A0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AIManager_TA");
}

return uClassPointer;
}

```

```

};

void eventDestroyed();
void RemoveBot(class AAIController_TA* Bot);
void AddBot(class AAIController_TA* Bot);
void Init(class AGameEvent_TA* InGameEvent);
void eventAILog(class UObject* Sender, class FString Message, unsigned long bVerbose);
};

// Class TAGame.AnimNodeBlend_TA
// 0x000F (0x0145 - 0x0154)
class UAnimNodeBlend_TA : public UAnimNodeBlendBase
{
public:
float BlendSpeed; // 0x0148 (0x0004)
[0x0000000000000001] (CPF_Edit)
float Position; // 0x014C (0x0004)
[0x0000000000002000] (CPF_Transient)
float DesiredPosition; // 0x0150 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNodeBlend_TA");
}

return uClassPointer;
};

};

// Class TAGame.AnimNodeBlendBoost_TA
// 0x0004 (0x0158 - 0x015C)
class UAnimNodeBlendBoost_TA : public UAnimNodeBlend
{
public:
float BlendTime; // 0x0158 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNodeBlendBoost_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.AnimNodeBlendSpeed_TA
// 0x000C (0x01A4 - 0x01B0)
class UAnimNodeBlendSpeed_TA : public UAnimNodeAimOffset
{
public:
float BlendRate; // 0x01A8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxSpeed; // 0x01AC (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNodeBlendSpeed_TA");
}

return uClassPointer;
};

};

// Class TAGame.AnimStateComponent_TA
// 0x002C (0x00A4 - 0x00D0)
class UAnimStateComponent_TA : public UActorComponent_X
{
public:
class USkeletalMeshComponent* OwnerSkeletalMeshComponent; // 0x00A8 (0x0008) [0x000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAnimNodeBlendList_X* BlendNode; // 0x00B0 (0x0008) [0x000004000000000]
struct FScriptDelegate _EventOnDetached_Delegate; // 0x00B8 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimStateComponent_TA");
}

return uClassPointer;
};

```

```

void eventDetached();
class UAnimNodeBlendList_X* GetBlendNode(class UAnimNode* ParentNode);
void RestartActiveSequence();
void eventAttached();
void EventOnDetached(class UAnimStateComponent_TA* DetachedComponent);
};

// Class TAGame.ArenaReflectionsManager_TA
// 0x0064 (0x0060 - 0x00C4)
class UArenaReflectionsManager_TA : public UObject
{
public:
class AGameEvent_TA*           SoccarGame;           // 0x0060 (0x0008)
[0x0000800000000000]
TArray<struct FSlapbackActorSetup>     SlapbackSetups;      // 0x0068
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FSlapbackActorSetup>     ArenaReverbSetups;    // 0x0078
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class AAkBusActor*>           SlapBackActors;       // 0x0088 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class AAkBusActor*>           ArenaReverbEmitters; // 0x0098
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FName                      ImpactIntensityKey;   // 0x00A8 (0x0008)
[0x0000000000000000]
struct FName                      SlapbackDistanceKey; // 0x00B0 (0x0008)
[0x0000000000000000]
struct FName                      ReverbBalanceKey;    // 0x00B8 (0x0008)
[0x0000000000000000]
float                            BallListenerOffset;  // 0x00C0 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ArenaReflectionsManager_TA");
}

return uClassPointer;
};

struct FVector eventGetPrimaryPlayerPosition();
void SetReverbBalance(float ReverbBalance);
void SetArenaReverbSettings(struct FName ArenaReverbShareset);
void SpawnBusActors(struct FVector& FieldSize);
float GetSlapbackDistance(struct FVector& ListenerLocation, struct FVector& ImpactLocation,
struct FVector& BusActorLocation);
void UpdateActorParams(float ImpactIntensity, struct FVector& ImpactLocation);
};

```

```

// Class TAGame.ArenaSoundSettings_TA
// 0x0108 (0x0060 - 0x0168)
class UArenaSoundSettings_TA : public UObject
{
public:
    class UAkSoundCue*           InGameMusicCue;          // 0x0060 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                 bDynamicMusic : 1;      // 0x0068 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long                 UpdateStatSounds : 1; // 0x0068 (0x0004)
    [0x0000000800000001] [0x00000002] (CPF_Edit)
    class UAkSoundCue*           GoalScoreSound;        // 0x0070 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           GameStartStinger;       // 0x0078 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           GoalScoreStinger;       // 0x0080 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           EpicSaveStinger;        // 0x0088 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           OvertimeStinger;        // 0x0090 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           WinStinger;             // 0x0098 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           LoseStinger;            // 0x00A0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           ForfeitStinger;         // 0x00A8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float                        GoalScoreStingerDelay; // 0x00B0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           Win;                   // 0x00B8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           Lose;                  // 0x00C0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           CountdownStart;        // 0x00C8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           CountdownEnd;          // 0x00D0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           OneMinRemaining;       // 0x00D8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           ThirtySecondsRemaining; // 0x00E0
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           OvertimeStart;         // 0x00E8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    TArray<struct FArenaStatSound> StatSounds;          // 0x00F0 (0x0010)
    [0x0000000000400041] (CPF_Edit | CPF_EditConstArray | CPF_NeedCtorLink)
    TArray<class UAkSoundCue*>   CountDownSoundList;     // 0x0100
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<class UAkSoundCue*>   RoundCountDownSoundList; // 0x0110
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    struct FAkEnvironment        GoalStingerEnvironment; // 0x0120 (0x0020)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    struct FAkEnvironment        LevelDefaultEnvironment; // 0x0140
    (0x0020) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    struct FName                  ArenaReverbShareset;   // 0x0160 (0x0008)

```

[0x0000000000000001] (CPF\_Edit)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ArenaSoundSettings_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.AttachmentBehavior_TA  
// 0x0014 (0x0060 - 0x0074)  
class UAttachmentBehavior_TA : public UObject  
{  
public:  
class UProductAsset_TA* Asset; // 0x0060 (0x0008)  
[0x0000004000002000] (CPF_Transient)  
class UActorComponent* Component; // 0x0068 (0x0008)  
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)  
unsigned long bPreview : 1; // 0x0070 (0x0004)  
[0x0000004000002000] [0x00000001] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.AttachmentBehavior_TA");  
}  
  
return uClassPointer;  
};  
  
class UCarMeshComponent_TA* GetMesh();  
void StopBehavior();  
void SetPreviewing(unsigned long bPreviewing);  
void OnInit();  
void Init(class UProductAsset_TA* InAsset, class UActorComponent* InComponent);  
};  
  
// Class TAGame.AttachmentBehavior_Steering_TA  
// 0x0044 (0x0074 - 0x00B8)  
class UAttachmentBehavior_Steering_TA : public UAttachmentBehavior_TA  
{  
public:
```

```

struct FName           BoneControlName;          // 0x0078 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAnimTree*      AnimTree;                // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat   SteeringToDegreesCurve;    // 0x0088
(0x0018) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class USkeletalMeshComponent*   SkelMeshComponent;    // 0x00A0
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | 
CPF_EditInline)
class USkelControlSingleBone*   BoneController;       // 0x00A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class AVehicle_TA*          Vehicle;               // 0x00B0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AttachmentBehavior_Steering_TA");
}

return uClassPointer;
};

void HandleAttached(class UCarMeshComponent_TA* Mesh);
void OnInit();
};

// Class TAGame.RimSpinner_TA
// 0x0050 (0x0074 - 0x00C4)
class URimSpinner_TA : public UAttachmentBehavior_TA
{
public:
float              AccelRate;                // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float              DecelRate;                // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
float              MaxSpeed;                 // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FName           BoneControlName;          // 0x0084 (0x0008)
[0x0000000000000001] (CPF_Edit)
float              SpinScale;                // 0x008C (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t             WheelIndex;               // 0x0090 (0x0004)
[0x0000000000002000] (CPF_Transient)
float              CurrentSpeed;              // 0x0094 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t             CurrentRotation;        // 0x0098 (0x0004)
[0x0000000000002000] (CPF_Transient)
class USkeletalMeshComponent*   SkelMeshComponent;    // 0x00A0
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | 
CPF_EditInline)

```

```

CPF>EditInline)
class USkelControlSingleBone*           SpinBoneController;          // 0x00A8
(0x0008) [0x0000000000002000] (CPF_Transient)
class USkelControlSingleBone*           WheelBoneController;        // 0x00B0
(0x0008) [0x0000000000002000] (CPF_Transient)
class UWheel_TA*                     Wheel;                  // 0x00B8 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
unsigned long                         bIsMainMenu : 1;           // 0x00C0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RimSpinner_TA");
}

return uClassPointer;
};

void StopBehavior();
void HandleAttached(class UCarMeshComponent_TA* Mesh);
void OnInit();
};

// Class TAGame.WheelShimmy_TA
// 0x0044 (0x0074 - 0x00B8)
class UWheelShimmy_TA : public UAttachmentBehavior_TA
{
public:
float                      AccumDistance;           // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      Wave1Freq;                // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      Wave2Freq;                // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      OnOffBias;               // 0x0084 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      HighFreq;                 // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      ShimmyMag;                // 0x008C (0x0004)
[0x000000004000001] (CPF_Edit | CPF_EditInline)
TArray<int32_t>            AffectedWheels;           // 0x0090 (0x0010)
[0x00000000400001] (CPF_Edit | CPF_NeedCtorLink)
class USkeletalMeshComponent*   SkelMeshComponent;       // 0x00A0
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UWheel_TA*             Wheel;                  // 0x00A8 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class USkelControlSingleBone* RollControl;           // 0x00B0 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelShimmy_TA");
}

return uClassPointer;
};

void OnInit();

// Class TAGame.PlayerControllerBase_TA
// 0x0108 (0x0878 - 0x0980)
class APlayerControllerBase_TA : public APlayerController_X
{
public:
float SpectatorCameraAccel; // 0x0878 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector SpectatorVelXY; // 0x087C (0x000C)
[0x0000000000002000] (CPF_Transient)
float SpectatorVelZ; // 0x0888 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UPostProcessManager_TA* PostProcessManagerArchetype; //
0x0890 (0x0008) [0x0000004000000001] (CPF_Edit)
class UPostProcessManager_TA* PostProcessManager; // 0x0898
(0x0008) [0x0000004000002000] (CPF_Transient)
int32_t SpectatorPitchDir; // 0x08A0 (0x0004)
[0x0000008000002000] (CPF_Transient)
int32_t SwivelPitchDir; // 0x08A4 (0x0004)
[0x0000008000002000] (CPF_Transient)
class ASpectatorVolume_TA* SpectatorVolume; // 0x08A8
(0x0008) [0x0000004000002000] (CPF_Transient)
float FullRotationScalar; // 0x08B0 (0x0004)
[0x0000000000000002] (CPF_Const)
float FullRotationMax; // 0x08B4 (0x0004)
[0x0000000000000002] (CPF_Const)
class UVanitySetManager_TA* VanityMgr; // 0x08B8 (0x0008)
[0x0000800000000000]
unsigned long bVanityInitialized : 1; // 0x08C0 (0x0004)
[0x0000000000000000] [0x00000001]
class UPlayerBanner_TA* PlayerBanner; // 0x08C8 (0x0008)
[0x0008000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPlayerAvatar_TA* PlayerAvatar; // 0x08D0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPlayerAvatarBorder_TA* PlayerBorder; // 0x08D8 (0x0008)
[0x0008000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAudioSpectatorMixComponent_TA* AudioSpectatorMixComponent; //
0x08E0 (0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

```

```
class UAudioFieldSideComponent_TA*           AudioFieldSideComponent;          // 0x08E8 (0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FScriptDelegate                     __EventLoadingUnfairMatch__Delegate;      // 0x08F0 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate                     __EventRemoveSSPlayer__Delegate;        // 0x0908 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate                     __EventReceivedCurrencyDrop__Delegate;   // 0x0920 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate                     __EventPlayerInputSet__Delegate;       // 0x0938 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate                     __PlayerBanner__ChangeNotify;         // 0x0950 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate                     __PlayerBorder__ChangeNotify;         // 0x0968 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerControllerBase_TA");
}

return uClassPointer;
};

void ResetView();
void QueryCrosshair();
class AAIController_TA* GetLockedDebugAI();
struct FVector GetAimLocation();
void ContextualDebugAction();
struct FUniqueNetId __PlayerControllerBase_TA__Say_TA_0x6(class APlayerReplicationInfo* P);
bool __PlayerControllerBase_TA__Say_TA_0x5(class APlayerReplicationInfo* P);
struct FUniqueNetId __PlayerControllerBase_TA__Say_TA_0x2(class APlayerReplicationInfo* P);
bool __PlayerControllerBase_TA__Say_TA_0x1(class APlayerReplicationInfo* P);
void __PlayerBorder__ChangeNotifyFunc();
void __PlayerBanner__ChangeNotifyFunc();
struct FVector UpdateSpectatorLocation(struct FVector CurrentLocation, float DeltaTime);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void ReceiveCurrencyDrop(struct FCurrency Drop);
bool FindSpectatorVolumeLocation(struct FVector CurrentLocation, struct FVector NextLocation,
struct FVector& NewLocation);
struct FVector ClampSpectatorLocation(struct FVector NewLocation);
void NotifyOnPlayerInputSet(struct FScriptDelegate Callback);
void SetPlayerInput(class UClass* NewInputClass);
void StartLanMatch(struct FCustomMatchSettings& Settings);
static class FString BuildMatchOptionsLan(struct FCustomMatchSettings& Settings);
void StartSplitscreenMatch(struct FCustomMatchSettings& Settings);
static void BuildMatchOptionsForPlaylist(class UGameSettingPlaylist_X* Playlist, TArray<class FString>& OptionsArray);
void StartMatch(struct FName PlaylistName, class FString URL);
static class FString BuildMatchOptions(class UGameSettingPlaylist_X* Playlist, TArray<class
```

```
FString> Options, struct FCustomMatchSettings& Settings);
static class FString JoinOption(class FString FullString, class FString Value);
static struct FName GetValidMapName(struct FName InName, int32_t GameMode);
void ClientRemoveSSPlayer(class FString Reason, class FString Title);
void RemoveSSPlayer(class FString Reason, class FString Title);
void OnGameInviteAccepted(class FString ErrorString, struct FOnlineGameSearchResult&
InviteResult);
void UpdateSpectatorViewRotation(float DeltaTime);
struct FRotator GetSpectatorViewRotationDeltaRotation(float DeltaTime);
void AddVoiceChatNotification(class FString NotificationMessage, uint8_t ChatChannel);
void EraseChatByType(uint8_t ChatChannel, struct FUniqueNetId Personald);
bool PartyLobbyMessage_TA(struct FUniqueNetId InSenderId, class FString PlayerName, class
FString Message, unsigned long bIsLocalPlayer);
void Say_TA(class FString Message, uint8_t ChatChannel, struct FUniqueNetId Recipient,
unsigned long bPreset, unsigned long bFriendsOnly);
void ServerSay_TA(class FString Message, uint8_t ChatChannel, struct FPlayerIdArray
ForbiddenPlayers, unsigned long bPreset);
void RepopulateChatHistory(uint8_t ChatChannel, struct FUniqueNetId Personald);
bool InviteHasEnoughSpace(class UOnlineGameSettings* InviteSettings);
void HandleReadBorder(TArray<struct FUniqueNetId> PlayerIds);
void HandleReadAvatar(TArray<struct FUniqueNetId> PlayerIds);
void HandleReadBanner(TArray<struct FUniqueNetId> PlayerIds);
void UpdateVanities(unsigned long bForceUpdate);
void InitVanityUI(class UProfile_TA* Profile, class UGFxShell_X* Shell);
void InitFromProfile(class UProfile_TA* Profile);
void HandleLoginStatusChanged(class UOnlinePlayer_X* InPlayer);
class UObject* GetProfileObject(class UClass* saveClass);
class UObject* GetSaveObject(class UClass* saveClass);
void OnExternalUIChanged(unsigned long bIsOpening);
class USaveData_TA* GetSaveData();
class UProfile_TA* GetProfile();
class ULocalPlayer_TA* GetLocalPlayer();
void UpdateForceFeedback();
void HandleForceFeedback(class UForceFeedbackManager* _);
void HandleProfileControlsSave(class UProfileControlsSave_TA* ControlsSave);
void UpdateAAType();
void HandlePostProcessManager(class UPostProcessManager_TA* PPM);
void HandleVideoSettingsSavePC(class UVideoSettingsSavePC_TA* VideoSavePC);
void UpdateCameraShake();
void HandleCamera(class ACamera_X* Camera);
void HandleCameraSave(class UProfileCameraSave_TA* CameraSettings);
void OnReceivedPlayerAndPRI();
void eventDestroyed();
void eventReceivedPlayer();
void eventPostBeginPlay();
void EventPlayerInputSet();
void EventReceivedCurrencyDrop(class APlayerControllerBase_TA* PC, struct FCurrency Drop);
void EventRemoveSSPlayer(class APlayerControllerBase_TA* PC, class FString Reason);
void EventLoadingUnfairMatch(class APlayerControllerBase_TA* PlayerControllerBase);
};

// Class TAGame.PlayerController_TA
// 0x03808 (0x0980 - 0x0D008)
class APlayerController_TA : public APlayerControllerBase_TA
```

```

{
public:
class ACar_TA*           Car;                      // 0x0980 (0x0008)
[0x0000008000002000] (CPF_Transient)
class APRI_TA*            PRI;                     // 0x0988 (0x0008)
[0x00000048000002000] (CPF_Transient)
struct FVehicleInputs      VehicleInput;           // 0x0990 (0x0020)
[0x0000000000002000] (CPF_Transient)
unsigned long              bReceivedServerShutdownMessage : 1;    // 0x09B0
(0x0004) [0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long              bFilterNonTacticalQuickChat : 1;    // 0x09B0 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long              bOverrideInput : 1;        // 0x09B0 (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long              bJumpPressed : 1;        // 0x09B0 (0x0004)
[0x0000000000000000] [0x00000008]
unsigned long              bBoostPressed : 1;        // 0x09B0 (0x0004)
[0x0000000000000000] [0x00000010]
unsigned long              bHandbrakePressed : 1;    // 0x09B0 (0x0004)
[0x0000000000000000] [0x00000020]
unsigned long              bHasPitchedOrRolled : 1;   // 0x09B0 (0x0004)
[0x0000000000000000] [0x00000040]
unsigned long              bAirPitchSafetyEnabled : 1; // 0x09B0 (0x0004)
[0x0000000000000000] [0x00000080]
unsigned long              bAllowAsymmetricalMute : 1; // 0x09B0 (0x0004)
[0x0000000000004000] [0x00000100] (CPF_Config)
unsigned long              bReportedPlayer : 1;       // 0x09B0 (0x0004)
[0x0008004000000000] [0x00000200]
class FString             LoginURL;                // 0x09B8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
uint8_t                   DeprecatedVoiceFilter;     // 0x09C8 (0x0001)
[0x0000000000002000] (CPF_Transient)
uint8_t                   QuickChatFilter;          // 0x09C9 (0x0001)
[0x0000000000002000] (CPF_Transient)
uint8_t                   MatchChatFilter;          // 0x09CA (0x0001)
[0x0000000000002000] (CPF_Transient)
struct FChatSpamData      ChatSpam;                 // 0x09CC (0x0018)
[0x0000000000000001] (CPF_Edit)
struct FChatSpamData      ChatSpamHarsh;           // 0x09E4 (0x0018)
[0x0000000000000000]
class ULightBarComponent_TA* LightBar;               // 0x0A00 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UGameMusicComponent_TA* MusicComponent;        // 0x0A08
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAudioPriorityComponent_TA* AudioPriorityComponent; // 0x0A10
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAudioMixStateComponent_TA* AudioMixStateComponent; //
0x0A18 (0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UTeamDemoAudioComponent_TA* TeamDemoAudioComponent; //
0x0A20 (0x0008) [0x00100004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class APRI_TA*             FollowTarget;            // 0x0A20 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ACamera*              SpectatorCameraArchetype; // 0x0A2830
(0x0008) [0x0000000000000001] (CPF>Edit)

```

```

class AHUD*           SpectatorHUDArchetype;          // 0x0A308 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UClass*         SpectatorInputClass;          // 0x0A3840 (0x0008)
[0x0000000000000001] (CPF_Edit)
class ACamera*        EditorCameraArchetype;        // 0x0A408 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AHUD*           EditorHUDArchetype;          // 0x0A4850 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UClass*         EditorInputClass;            // 0x0A508 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UIInterface_GameEditor_TA* MoveActor_Object;    // 0x0A5860
(0x0008) [0x0000000000000000]
class UIInterface_GameEditor_TA* MoveActor_Interface; // 0x0A608
(0x0008) [0x0000000000000000]
struct FVector        MoveActorGrabOffset;         // 0x0A6870 (0x000C)
[0x0000000000000000]
float                 MoveActorGrabIncrement;       // 0x0A74C (0x0004)
[0x0000000000000000]
float                 MinMoveActorGrabDistance;     // 0x0A780 (0x0004)
[0x0000000000000000]
float                 MouseIncrementSpeed;          // 0x0A7C84 (0x0004)
[0x0000000000000000]
float                 BallVelocityIncrementAmount;   // 0x0A808 (0x0004)
[0x0000000000000000]
int32_t               BallVelocityIncrementFireCount; // 0x0A84C (0x0004)
[0x0000000000000000]
float                 BallVelocityIncrementFireCountMax; // 0x0A8890 (0x0004)
[0x0000000000000000]
float                 BallVelocityIncrementSpeedDefault; // 0x0A8C94 (0x0004)
[0x0000000000000000]
float                 BallVelocityIncrementSpeedMax;   // 0x0A908 (0x0004)
[0x0000000000000000]
float                 CrosshairTraceDistance;        // 0x0A94C (0x0004)
[0x0000000000000000]
class AActor*          TracedCrosshairActor;        // 0x0A98A0 (0x0008)
[0x0000000000000000]
TArray<struct FCrosshairExtentInfo> CrosshairTraceExtents; // 0x0AA08
(0x0010) [0x00000000040000] (CPF_NeedCtorLink)
class UIInterface_GameEditor_TA* RotatedActor_Object;    // 0x0AB08
(0x0008) [0x0000000000000000]
class UIInterface_GameEditor_TA* RotatedActor_Interface; // 0x0AB8C0
(0x0008) [0x0000000000000000]
struct FVector        RotateActorCameraLocationOffset; // 0x0AC08
(0x000C) [0x0000000000000000]
struct FVector        RotateActorCameraRotationOffset; // 0x0ACCD4
(0x000C) [0x0000000000000000]
int32_t               RotateActorCameraSide;          // 0x0AD8E0 (0x0004)
[0x0000000000000000]
float                 DesiredCameraSide;            // 0x0ADCE4 (0x0004)
[0x0000000000000000]
float                 PawnTypeChangedTime;          // 0x0AE08 (0x0004)
[0x0000000000000000]
int32_t               SelectedSpawnArchetype;        // 0x0AE4C (0x0004)
[0x0000000000000000]

```

```

struct FVehicleInputsOverrideInput; // 0x0AE8F0 (0x0020)
[0x0000000000002000] (CPF_Transient)
int32_t MinClientInputRate; // 0x0B108 (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t MedianClientInputRate; // 0x0B0C14 (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t MaxClientInputRate; // 0x0B108 (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t ConfiguredClientInputRate; // 0x0B14C (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
float TimeSinceLastMovePacket; // 0x0B1820 (0x0004)
[0x0000000000002000] (CPF_Transient)
float TimeLastReplicatedMovePacket; // 0x0B1C24 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UPlatformMetrics_TA* PlatformMetrics; // 0x0B208
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float MouseXDeadZone; // 0x0B2830 (0x0004)
[0x0000000000000000]
float MouseYDeadZone; // 0x0B2C34 (0x0004)
[0x0000000000000000]
float MouseXDeadZoneAir; // 0x0B308 (0x0004)
[0x0000000000000000]
float MouseYDeadZoneAir; // 0x0B34C (0x0004)
[0x0000000000000000]
struct FVehicleInputsLastInputs; // 0x0B3840 (0x0020)
[0x0000000000000000]
class APRI_TA* PendingViewPRI; // 0x0B5860 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UClientConnectionTracker_TA* ConnectionTracker; // 0x0B608
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UProfileCameraSave_TA* CameraSave; // 0x0B6870
(0x0008) [0x0000000000002000] (CPF_Transient)
class UEngineShare_TA* EngineShare; // 0x0B708 (0x0008)
[0x0000800000000000]
class ANetworkInputBuffer_TA* InputBuffer; // 0x0B780 (0x0008)
[0x0000000100000020] (CPF_Net)
class UCrossplayConfig_X* CrossplayConfig; // 0x0B808 (0x0008)
[0x0000800000000000]
class FString PlayerReportedMessage; // 0x0B8890 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString PlayerReportedPostGameMessage; // 0x0B98A0
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString CheckPlayerReportStatusMessage; // 0x0BA8B0
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FVoiceRoomCredentials RoomCredentials; // 0x0BB8C0
(0x0030) [0x000100000400000] (CPF_NeedCtorLink)
class UEOSGameClipsController_TA* EOSGameClipsControllerArchetype; //
0x0BE8F0 (0x0008) [0x0009000000000000]
class UEOSGameClipsController_TA* EOSGameClipsController; //
0x0BF08 (0x0008) [0x0009000000000000]
struct FScriptDelegate __EventLaunchAccountPicker__Delegate; //
0x0BF8C00 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```
struct FScriptDelegate           __EventLaunchControllerApplet__Delegate;    // 0x0C108
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventMuteChanged__Delegate;            // 0x0C2830
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventTrainingEditorActorModified__Delegate; // 0x0C408 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSelectCameraTarget__Delegate;     // 0x0C5860
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventConnectionTrackerAttached__Delegate; // 0x0C708 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventChatMessage__Delegate;          // 0x0C8890
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventOverrideInput__Delegate;        // 0x0CA08
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bReportedPlayer__ChangeNotify;      // 0x0CB8C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EOSGameClipsControllerArchetype__ChangeNotify;// 0x0CD08 (0x0018) [0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EOSGameClipsController__ChangeNotify;   // 0x0CE8F0 (0x0018) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerController_TA");
}

return uClassPointer;
};

void HandleCarSet(class APRI_TA* InPRI);
void SpawnSelectedArchetype();
bool RemoveActor();
void ToggleGrabActor();
void ToggleRotateActor();
void EditorReleaseActor();
void EditorCycleActor();
void DuplicateShot();
void StopEditing();
void EditorIncreasePower();
void EditorDecreasePower();
void EditorIncreasePowerToggleInterim();
void EditorDecreasePowerToggleInterim();
void EditorIncreasePowerToggle(unsigned long bToggle);
void EditorDecreasePowerToggle(unsigned long bToggle);
void ModifyEditorPower(int32_t Direction);
void ToggleCameraPosition();
void EditorUndo();
void EditorRedo();
void EditorIncreaseRoundTime();
```

```
void EditorDecreaseRoundTime();
void EditorNextRound();
void EditorPrevRound();
void UpdateCrosshair();
void __PlayerController_TA__ReceivedPlayer_0x1(class UBanSync_TA* BanSync);
void __EOSGameClipsController__ChangeNotifyFunc();
void __EOSGameClipsControllerArchetype__ChangeNotifyFunc();
void __bReportedPlayer__ChangeNotifyFunc();
void NetClientInputRate(int32_t Rate);
void ClientStayAsPartyVoteBegin();
void ServerUpdateCustomMatchSettings(struct FCustomMatchSettings Settings);
void ClientUpdateTournamentMatch(struct FTourMatch Match, int32_t GameNum, struct FTourMatchGame GameData);
void ServerCreateMatchBroadcast(class AGameEvent_Soccar_TA* GameEvent);
void ClampMoveActorGrabOffset();
void RevertToDefaultCameraHUDInput();
void SetCameraHUDInput(class ACamera* InCameraArchetype, class AHUD* InHUDArchetype, class UClass* InPlayerInputClass);
void SwitchToEditPawn();
void ToggleEditorRound();
void ToggleBetweenCarAndEditPawn();
void Interact();
void StopMovement(unsigned long bOnlyIfNoAccel);
struct FVector GetRotateActorCameraOffset(float DeltaTime, unsigned long bSnap);
void RestoreEditorPawnOrientation();
void BackupEditorPawnOrientation();
bool EditorPawnDeSerialize(class UJsonObject* Data);
class FString EditorPawnSerialize();
void UpdateRotatedActorOrientation(float DeltaTime);
void OnOpenPauseMenu();
void ResetMouseCenter();
float CalculateMouseAxis(float Center, float CurrentLocation, float Deadzone, float MaxDist);
struct FVector CalculateMouseAccelInput(float XDeadZone, float YDeadZone, float XMaxDist, float YMaxDist, struct FVector& CurrentLocation);
void ShowControllerApplet();
void ShowAccountPicker();
void QueSaveReplay();
void SetFollowTarget(class APRI_TA* InTarget);
void FollowPlayer(class APRI_TA* InPlayer);
void ClientSkillUpdated(struct FUpdatedPlayerSkillRating Rating, int32_t Playlist);
void ServerReportPlayer(uint8_t ReasonIDStaticArray, int32_t NumReasons, struct FUniqueNetId PlayerID);
void ReportPlayer(TArray<uint8_t> ReasonIDs, struct FUniqueNetId PlayerID);
void HandleLocalStatEvent(class APRI_TA* InPRI, class UStatEvent_TA* StatEvent, int32_t Count);
void ClientSplitscreenJoinResponse(struct FUniqueNetId PlayerID, unsigned long bAllow, class FString Error);
void ServerRequestSplitscreenJoin(struct FUniqueNetId PlayerID, class FString PlayerName);
bool eventPreClientTravel(class FString PendingURL, uint8_t TravelType, unsigned long bIsSeamlessTravel);
void NotifyGoalScored(int32_t ScoredOnTeam);
void eventDestroyed();
void SetGameHUDArchetype(class AHUD* InArchetype);
bool ShouldBeMuted(class APlayerController_TA* Other);
```

```
bool IsCommunicationFiltered(uint8_t Filter, class APlayerController_TA* Other, unsigned long bPreset, unsigned long bTactical, unsigned long bFilterNonTactical);
void RefreshMutedPlayers(unsigned long bForceRefresh);
void eventServerUnmutePlayer(struct FUniqueNetId PlayerNetId);
void eventServerMutePlayer(struct FUniqueNetId PlayerNetId);
void GameplayUnmutePlayer(struct FUniqueNetId PlayerNetId);
void GameplayMutePlayer(struct FUniqueNetId PlayerNetId);
class FString GetDebugMutePlayerName(struct FUniqueNetId PlayerNetId);
void ServerSetMatchChatFilter(uint8_t InMatchChatFilter);
void ServerSetTacticalChatFilter(unsigned long bInFilterNonTacticalQuickChat);
void ServerSetQuickChatFilter(uint8_t InQuickChatFilter);
void ServerSetVoiceChatFilter(uint8_t InVoiceChatFilter);
bool IsExplicitlyMuted(struct FUniqueNetId PlayerNetId);
void DebugAI();
void SendPendingRPCs();
bool ApplySocialBanFilter(unsigned long bQuickChatMessage);
bool ApplyChatBanFilter(unsigned long bQuickChatMessage);
bool ApplyChatSpamFilter(struct FChatSpamData& OutChatSpam);
bool CanSendMessageW(unsigned long bQuickChatMessage);
void ClientNotifyChatBannedPermanently();
void ClientNotifyChatBanned(uint64_t ChatBanExpiration);
void ClientNotifyChatDisabled(float Time);
void ChatMessage_TA(class APlayerReplicationInfo* InPRI, class FString Message, uint8_t ChatChannel, unsigned long bPreset, int32_t TimeStamp);
void ServerSayInternal_TA(class FString Message, uint8_t ChatChannel, TArray<struct FUniqueNetId> ForbiddenPlayers, unsigned long bPreset);
void ServerSay_TA(class FString Message, uint8_t ChatChannel, struct FPlayerIdArray ForbiddenPlayers, unsigned long bPreset);
void Say_TA(class FString Message, uint8_t ChatChannel, struct FUniqueNetId Recipient, unsigned long bPreset, unsigned long bFriendsOnly);
bool CheckMatchChatOrQuickChat(unsigned long bPreset, uint8_t QuickChatTest, uint8_t MatchChatTest);
bool CanChatWith(class APlayerController_TA* Other, unsigned long bPreset, unsigned long bTactical);
bool AllowTextMessage(class FString msg);
void PushToTalkEnd();
void PushToTalk();
bool ShouldTournamentMatchmake();
void HandleJoinGameMigrationCompleted(unsigned long bSuccess, class FString FailReason);
void ClientNotifyServerShutdown(struct FServerConnectionInfo ConnectionInfo);
void eventClientUnmutePlayer(struct FUniqueNetId PlayerNetId);
void eventClientMutePlayer(struct FUniqueNetId PlayerNetId, unsigned long bAddToMuteList);
void GetOnlineStatus(unsigned long bLocalize, class FString& PresenceString, class FString& GameDataString);
void ClientSetOnlineStatus();
void HandleGameDataSelected(int32_t PlaylistId, int32_t MutatorIndex);
void PrintDebugInfo(class UDebugDrawer* Drawer);
class AGameEvent_TA* GetGameEvent();
void ClientArbitratedMatchEnded();
void HandleDisconnected(class UPsyNetConnection_X* Connection);
void BannedKick();
void NoReservationKick();
void IdleKick();
void ServerReportServer();
```

```
void ReportServer();
void ClientUseItem(struct FVector UseLocation, struct FRotator UseRotation);
void ServerTeleportCar(struct FVector SpawnLocation, struct FRotator NewRotation);
void eventTeleportCar(struct FVector SpawnLocation, struct FRotator NewRotation);
void ReceiveMessage(struct FMessagePacket Packet);
void GameEventCommand(class FString CommandType);
void NextPickup();
void ServerUsePickupReleased();
void UsePickupReleased();
void ServerUsePickup(class ASpecialPickup_TA* ActivatedPickup, class ARBActor_TA* Target);
void UsePickup();
void ButtonMash(unsigned long bMash);
void Grab(unsigned long bGrab);
void ToggleHandbrake(unsigned long bHandbrake);
void ToggleBoost(unsigned long bBoost);
void ToggleJump(unsigned long bJump);
void TargetSelectLeft();
void TargetSelectRight();
void ReleaseRearCamera();
void PressRearCamera();
void ReleaseSecondaryCamera();
void PressSecondaryCamera();
void CaptureClip(class FString InGameClipInputType);
void ReadyUp();
void Spectate();
void ChangeTeam(int32_t TeamNum);
void SwitchTeam();
void SetDefaultCameraMode();
void ResetCameraMode();
struct FName GetCameraMode();
void SetCameraMode(struct FName NewCamMode);
void ProcessMove_TA(struct FVehicleInputs& NewInput);
void ZeroAllInput();
void ZeroMoveInput();
void AddMouseMovementVal(float PositiveInputVal, float NegativeInputVal, float InputDirVal,
unsigned long bUsingMouseForPositiveAxis, unsigned long bUsingMouseForNegativeAxis, float&
VehicleInputVal, float& CarAirAccelVal);
void AddInputVal(float InputVal, unsigned long bAddInputCondition, float& VehicleInputVal);
void ApplyAirPitchSafety();
void ModifiedAirRotationByMouseInput(class UPlayerInput_TA* InputTA, float DeadZoneX, float
DeadZoneY);
bool UpdateMouseGroundSteer(class UPlayerInput_TA* InputTA, float DeltaTime, float
Deadzone, float MaxDist, float BrakingSpeed);
bool UpdateMouseGroundThrottle(class UPlayerInput_TA* InputTA, float DeltaTime, float
Deadzone, float MaxDist);
void PlayerMove(float DeltaTime);
void IgnoreMoveInput(unsigned long bNewMoveInput);
void eventSendClientAdjustment();
class APlayerController_TA* GetPrimaryPlayerController();
void HandleAddBoostComponent(class ACarComponent_Boost_TA* Boost);
void OnPawnChange(class APawn* OldPawn, class APawn* NewPawn);
void HandleSetProfile(class ULocalPlayer_TA* LocalPlayer);
class UProductMetrics_TA* GetProductMetrics();
void ReplicateLoadout();
```

```
void ReplicateCameraRotation();
void SetUsingFreecam(unsigned long bFreecam);
void SetUsingBehindView(unsigned long bBehindView);
void SetUsingSecondaryCamera(unsigned long bSecondaryCamera);
void ReplicateCameraSettings();
void HandleControllerLayoutChanged(class APRI_TA* InPRI);
void HandlePawnTypeChanged(class APRI_TA* InPRI);
void HandlePendingViewCarSet(class APRI_TA* InPRI);
void HandleReplaceBot(class APRI_TA* InPRI);
void ClientNotifyVoiceBan();
void ClientNotifyVoiceRoomError(struct FName VoiceErrorName);
void HandleRoomCredentials();
void HandleMatchToken(class UPsyNetService_VoiceChatMatchToken_TA* Service);
void ClientNotifyVoiceTokenSecondHalf(class FString SecondHalfToken);
void ClientNotifyVoiceTokenFirstHalf(class FString FirstHalfToken);
void ClientNotifyVoiceBaseUrl(class FString BaseUrl);
void ClientNotifyVoiceRoomID(class FString RoomId);
void ServerRequestVoiceRoomToken(unsigned long bClearCache);
class UBanMessage_X* GetBanMessage(uint8_t BanType);
bool GetIsBanned(uint8_t BanType);
class FString GetMatchVoiceRoomName();
bool CurrentPlaylistAllowsVoiceChat();
void RequestVoiceRoomToken(unsigned long bClearCache);
void HandleTeamChanged(class APRI_X* InPRI);
void HandleMatchEnded(class AGameEvent_Soccar_TA* GameEvent);
void HandleGameEventChanged(class APRI_TA* InPRI);
void HandlePRICameraChanged(class APRI_TA* InPRI);
void SetupLightBar();
void HandlePersistentCameraSet(class APRI_TA* InPRI);
void OnReceivedPlayerAndPRI();
void OnSettingsAutoUpdated(uint8_t Reason);
void HandleProfileGamepadSave(class UProfileGamepadSave_TA* GamepadSave);
void HandleProfilePCSave(class UProfilePCSave_TA* ProfilePCSave);
void UpdateVoiceChatFilter();
void HandleSoundSettingsSave(class USoundSettingsSave_TA* SoundSettings);
void HandleMatchChatFilterChanged();
void HandleGameplaySettingsSave(class UGameplaySettingsSave_TA* GameplaySettings);
void HandleCameraSave(class UProfileCameraSave_TA* CameraSettings);
void ClientSetLevelSessionID(struct FGuid Id);
void ReplicateLevelSessionID();
void SetInputBuffer(class ANetworkInputBuffer_TA* InBuffer);
class UClass* GetInputBufferClass(uint8_t Type);
void ServerInitInputBuffer(uint8_t Type);
void NetworkInputBufferChanged(class UNetworkSave_TA* NetworkSave);
void InitNetworkSave(class UNetworkSave_TA* NetworkSave);
void InitEOSGameClipsController();
void eventReceivedPlayer();
void InitFromGRI(class AGRI_X* GRI);
void HandleGRISpawned(class AGRI_X* GRI);
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName EventName);
void EventOverrideInput(class APlayerController_TA* PC);
void EventChatMessage(class APlayerController_TA* PC, class FString Message, unsigned long bPreset);
```

```

void EventConnectionTrackerAttached(class APlayerController_TA* PC, class
UClientConnectionTracker_TA* Tracker);
void EventSelectCameraTarget(class APlayerController_TA* PC, int32_t Direction);
void EventTrainingEditorActorModified();
void EventMuteChanged(class APlayerController_TA* PC, struct FUniqueNetId PlayerID, unsigned
long bMuted);
void EventLaunchControllerApplet();
void EventLaunchAccountPicker(int32_t ControllerId);
};

// Class TAGame.AudioPriorityComponent_TA
// 0x006C (0x00A4 - 0x0110)
class UAudioPriorityComponent_TA : public UActorComponent_X
{
public:
struct FInterpCurveFloat           CarDistanceCurve;           // 0x00A8 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat           BallDistanceCurve;         // 0x00C0 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float                             DrivingAtListenerDotThreshold; // 0x00D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t                           DrivingAtPlayerBonus;      // 0x00DC (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t                           BoostingAtPlayerBonus;     // 0x00E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FColor>            DebugColors;             // 0x00E8 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FPriorityCandidate> PriorityArray;          // 0x00F8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class ABall_TA*                   Ball;                    // 0x0108 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioPriorityComponent_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
struct FPriorityCandidate CreateCandidate(class AVehicle_TA* Vehicle, struct FVector
CenterLocation);
void Tick(float DeltaTime);
void UpdateBallReference(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* InBall);
void eventDetached();
void eventAttached();
};

```

```
// Class TAGame.AudioStateComponent_TA
// 0x0020 (0x0258 - 0x0278)
class UAudioStateComponent_TA : public UPrimitiveComponent
{
public:
class FString           StateGroup;          // 0x0258 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString           StateName;          // 0x0268 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioStateComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.BallLocationComponent_TA
// 0x000B (0x009D - 0x00A8)
class UBallLocationComponent_TA : public UActorComponent
{
public:
class UMaterialInstanceConstant*      MaterialToApplyBallLocationTo;    // 0x00A0
(0x0008) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallLocationComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.BallRadiusVolume_TA
// 0x0020 (0x0268 - 0x0288)
class ABallRadiusVolume_TA : public AActor
{
public:
float             Radius;            // 0x0268 (0x0004)
```

```

[0x0000000000000001] (CPF_Edit)
class UDrawSphereComponent* PreviewRadius; // 0x0270
(0x0008) [0x00000000408000A] (CPF_Const | CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<class ABall_TA*> InRadius; // 0x0278 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallRadiusVolume_TA");
}

return uClassPointer;
};

void RemoveBall(class ABall_TA* Ball);
void AddBall(class ABall_TA* Ball);
void eventTick(float DeltaTime);
};

// Class TAGame.RBActor_TA
// 0x0250 (0x0558 - 0x07A8)
class ARBActor_TA : public APawn_X
{
public:
struct FPointer VfTable_IITickNotify_TA; // 0x0558 (0x0008)
[0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
float MaxLinearSpeed; // 0x0560 (0x0004)
[0x0000000100000023] (CPF>Edit | CPF_Const | CPF_Net)
float MaxAngularSpeed; // 0x0564 (0x0004)
[0x0000000100000023] (CPF>Edit | CPF_Const | CPF_Net)
float RollingFriction; // 0x0568 (0x0004)
[0x0000000000000002] (CPF_Const)
unsigned long bDisableSleeping : 1; // 0x056C (0x0004)
[0x0000000000000003] [0x00000001] (CPF>Edit | CPF_Const)
unsigned long bReplayActor : 1; // 0x056C (0x0004)
[0x0000000000002022] [0x00000002] (CPF_Const | CPF_Net | CPF_Transient)
unsigned long bFrozen : 1; // 0x056C (0x0004)
[0x0000004100002020] [0x00000004] (CPF_Net | CPF_Transient)
unsigned long bAutoInitFXEffects : 1; // 0x056C (0x0004)
[0x0000000000000002] [0x00000008] (CPF_Const)
unsigned long bIgnoreSyncing : 1; // 0x056C (0x0004)
[0x0000000000002020] [0x00000010] (CPF_Net | CPF_Transient)
unsigned long bPhysInitialized : 1; // 0x056C (0x0004)
[0x0000004000002000] [0x00000020] (CPF_Transient)
unsigned long bSkipNextErrorAccumulation : 1; // 0x056C (0x0004)
[0x0000004000002000] [0x00000040] (CPF_Transient)
struct FNetworkSyncSettingsData NetworkSyncSettings; // 0x0570
(0x0020) [0x0000000000000001] (CPF>Edit)

```

```

struct FReplicatedRBState           OldRBState;          // 0x0590 (0x0040)
[0x0000000000000200] (CPF_Const | CPF_Transient)
struct FReplicatedRBState          RBState;            // 0x05D0 (0x0040)
[0x0000000000000200] (CPF_Const | CPF_Transient)
struct FReplicatedRBState          ReplicatedRBState; // 0x0610 (0x0040)
[0x0000000000000202] (CPF_Const | CPF_Net | CPF_Transient)
struct FReplicatedRBState          ClientCorrectionRBState; // 0x0650
(0x0040) [0x0000000000000200] (CPF_Const | CPF_Transient)
struct FWorldContactData           WorldContact;       // 0x0690 (0x0028)
[0x0000000000000200] (CPF_Const | CPF_Transient)
struct FVector                   SyncErrorLocation; // 0x06B8 (0x000C)
[0x0000000000000200] (CPF_Const | CPF_Transient)
float                           SyncErrorAngle;      // 0x06C4 (0x0004)
[0x0000000000000200] (CPF_Const | CPF_Transient)
struct FVector                   SyncErrorAxis;       // 0x06C8 (0x000C)
[0x0000000000000200] (CPF_Const | CPF_Transient)
class UAkParamGroup*             Ak;                  // 0x06D8 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class AFXActor_X*               FXActorArchetype;   // 0x06E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X*               FXActor;            // 0x06E8 (0x0008)
[0x0000008000002000] (CPF_Transient)
class UImpactEffectsComponent_TA* ImpactEffectsComponent; // 0x06F0 (0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<struct FAccumulatedRigidBodyCollision> RBCollisions; // 0x06F8
(0x0010) [0x0000000000482002] (CPF_Const | CPF_Transient | CPF_Component | CPF_NeedCtorLink)
class UClass*                   RBHistoryClass;     // 0x0708 (0x0008)
[0x0000000000000001] (CPF>Edit)
class URBHistory_TA*            RBHistory;          // 0x0710 (0x0008)
[0x000000000408200A] (CPF_Const | CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
int32_t                         LastRBCollisionsFrame; // 0x0718 (0x0004)
[0x0000000000000200] (CPF_Const | CPF_Transient)
class UReplayComponent_TA*      Replay;              // 0x0720 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FWeldingInfo              WeldedInfo;         // 0x0728 (0x0028)
[0x000000A100002020] (CPF_Net | CPF_Transient)
class ARBActor_TA*              WeldedActor;        // 0x0750 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ARBActor_TA*              WeldedTo;           // 0x0758 (0x0008)
[0x0000004000002000] (CPF_Transient)
float                           PreWeldMass;        // 0x0760 (0x0004)
[0x00000000000002000] (CPF_Transient)
float                           ReplicatedGravityScale; // 0x0764 (0x0004)
[0x0000000100002020] (CPF_Net | CPF_Transient)
float                           ReplicatedCollisionScale; // 0x0768 (0x0004)
[0x0000000100002020] (CPF_Net | CPF_Transient)
struct FPointer                 Constraint2D;       // 0x0770 (0x0008)
[0x00000000000003000] (CPF_Native | CPF_Transient)
struct FPointer                 BulletRigidBody;    // 0x0778 (0x0008)
[0x00000000000003000] (CPF_Native | CPF_Transient)
class UPhysicsConfig_TA*        PhysicsConfig;      // 0x0780 (0x0008)

```

```
[0x0000800000000000]
uint8_t TeleportCounter; // 0x0788 (0x0001)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FScriptDelegate __EventRigidBodyCollision__Delegate; // 0x0790
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RBActor_TA");
}

return uClassPointer;
};

void OnRBActorTeleport();
void SetTeleportCount(uint8_t InTeleportCounter);
void RegisterTeleport();
void OnWeldedToChanged();
void eventSetWeldedTo(class ARBActor_TA* RBActor);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void GetTimeOfImpact(class ARBActor_TA* HitActor, struct FTimeOfImpactData& OutData);
void SetMass(float NewMass);
void SetConstrained3D(struct FVector LinearLower, struct FVector LinearUpper, struct FVector
AngularLower, struct FVector AngularUpper);
void SetConstrained2D(unsigned long bConstrain2D);
void EnableCCD(unsigned long bEnable);
void SetPhysicsState(struct FReplicatedRBState NewState);
void SetGravityScale(float InGravityScale);
void SetCollisionScale(float InCollisionScale);
void SetCollisionScaleNative(float NewScale);
void SetFrozen(unsigned long bEnabled);
void SetDyingPhysics();
void SetMaxAngularSpeed(float NewMaxSpeed);
void SetMaxLinearSpeed(float NewMaxSpeed);
void eventOnRigidBodyCollision(struct FAccumulatedRigidBodyCollision Collision);
void AddTorque(struct FVector Torque, uint8_t ForceMode, class FString DebugSource);
void AddForce(struct FVector Force, uint8_t ForceMode, class FString DebugSource);
void eventPreAsyncTick(float DeltaTime);
void OnWeldedActorChanged();
void SetWeldedActor(struct FWeldingInfo Welding);
void SetWelding(struct FWeldingInfo Welding);
void UnWeldRBActor(class ARBActor_TA* Other);
void WeldRBActor(class ARBActor_TA* Other, struct FVector WeldOffset, struct FRotator
WeldRotation);
void eventOnRBPhysInit();
void RelnitRBPhys();
void TerminateRBPhys();
void SetCurrentRBState(struct FReplicatedRBState& NewRBState);
struct FVector GetCurrentRBLocation();
```



[0x0000000000000021] (CPF\_Edit | CPF\_Net)  
class UBallCamTarget\_TA\* BallCamTarget; // 0x0838 (0x0008)  
[0x000000004080008] (CPF\_ExportObject | CPF\_Component | CPF\_EditInline)  
float Radius; // 0x0840 (0x0004)  
[0x0000000000002000] (CPF\_Transient)  
float VisualRadius; // 0x0844 (0x0004)  
[0x0000000000000000]  
TArray<struct FBallHitInfo> Touches; // 0x0848 (0x0010)  
[0x0000000000402000] (CPF\_Transient | CPF\_NeedCtorLink)  
float LastCalculateCarHit; // 0x0858 (0x0004)  
[0x0000000000002000] (CPF\_Transient)  
struct FVector InitialLocation; // 0x085C (0x000C)  
[0x0000000000002000] (CPF\_Transient)  
struct FRotator InitialRotation; // 0x0868 (0x000C)  
[0x0000000000002000] (CPF\_Transient)  
float LastHitWorldTime; // 0x0874 (0x0004)  
[0x0000000000002000] (CPF\_Transient)  
float ReplicatedBallScale; // 0x0878 (0x0004)  
[0x0000000100000020] (CPF\_Net)  
class UStaticMesh\* ReplicatedBallMesh; // 0x0880 (0x0008)  
[0x0000000100000020] (CPF\_Net)  
float ReplicatedWorldBounceScale; // 0x0888 (0x0004)  
[0x0000000100000020] (CPF\_Net)  
float ReplicatedBallGravityScale; // 0x088C (0x0004)  
[0x0000000100000020] (CPF\_Net)  
float ReplicatedBallMaxLinearSpeedScale; // 0x0890 (0x0004)  
[0x0000000100000020] (CPF\_Net)  
float ReplicatedAddedCarBounceScale; // 0x0894 (0x0004)  
[0x0000000000000020] (CPF\_Net)  
float AdditionalCarGroundBounceScaleZ; // 0x0898 (0x0004)  
[0x0000000000000000]  
float AdditionalCarGroundBounceScaleXY; // 0x089C (0x0004)  
[0x0000000000000000]  
class UPhysicalMaterial\* ReplicatedPhysMatOverride; // 0x08A0  
(0x0008) [0x0000000100000020] (CPF\_Net)  
uint8\_t HitTeamNum; // 0x08A8 (0x0001)  
[0x000000410000200] (CPF\_Net | CPF\_Transient)  
class AGameEvent\_Soccar\_TA\* GameEvent; // 0x08B0  
(0x0008) [0x000000810000200] (CPF\_Net | CPF\_Transient)  
struct FExplosionData ReplicatedExplosionData; // 0x08B8 (0x0018)  
[0x000000410008200] (CPF\_Net | CPF\_Transient | CPF\_Component)  
struct FExplosionDataExtended ReplicatedExplosionDataExtended; // 0x08D0  
(0x0020) [0x000000410008200] (CPF\_Net | CPF\_Transient | CPF\_Component)  
class AExplosion\_X\* Explosion; // 0x08F0 (0x0008)  
[0x0000004000002000] (CPF\_Transient)  
float ExplosionTime; // 0x08F8 (0x0004)  
[0x0000004000002000] (CPF\_Transient)  
struct FVector OldLocation; // 0x08FC (0x000C)  
[0x0000004000002000] (CPF\_Transient)  
TArray<class UMaterialInterface\*> FadeMaterials; // 0x0908 (0x0010)  
[0x0000000000400000] (CPF\_NeedCtorLink)  
float PredictionTimestep; // 0x0918 (0x0004)  
[0x0000000000000001] (CPF>Edit)  
TArray<struct FPredictedPosition> PredictedPositions; // 0x0920

```

(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
float LastPredictionTime; // 0x0930 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
struct FGoalPenetrationData GoalPenetration; // 0x0938 (0x0028)
[0x0000000000082000] (CPF_Transient | CPF_Component)
float GroundForce; // 0x0960 (0x0004)
[0x0000000000000001] (CPF_Edit)
class ACar_TA* CurrentAffector; // 0x0968 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UBallTrajectoryComponent_TA* TrajectoryComponent; // 0x0970
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
class UPitchTekDrawingComponent_TA* PitchTekComponent; // 0x0978
(0x0008) [0x000000000408000A] (CPF_Const | CPF_ExportObject | CPF_Component | CPF>EditInline)
class AGoalExplosionOrientation_TA* GoalExplosionOrientation; // 0x0980
(0x0008) [0x000004000002000] (CPF_Transient)
struct FScriptDelegate __EventCarTouch__Delegate; // 0x0988
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventHitWorld__Delegate; // 0x09A0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventHitGround__Delegate; // 0x09B8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventHitGoal__Delegate; // 0x09D0 (0x0018)
[0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventGameEventSet__Delegate; // 0x09E8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventExploded__Delegate; // 0xA00
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventHitTeamNumChanged__Delegate; // 0xA18
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTraillIntensityChanged__Delegate; // 0xA30
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Ball_TA");
}

return uClassPointer;
};

void OnRBActorTeleport();
void DisableOwnerTranslucency();
void EnableOwnerTranslucency(class AActor* NewOwner);
void OnWeldedToChanged();
float eventGetTraillIntensity();
struct FLinearColor eventGetActiveTeamColor();
void PredictGroundPosition(int32_t MaxIterations, struct FBallPredictionInfo& PredictionInfo);
void PredictPosition(float Time, struct FBallPredictionInfo& PredictionInfo);

```

```
void SetPhysMatOverride(class UPhysicalMaterial* InPhysMatOverride);
void DisableTrajectory();
bool CanEverShowTrajectory();
bool IsTrajectoryEnabled();
struct FVector GetTrajectoryStartVelocity();
struct FRotator GetTrajectoryStartRotation();
struct FVector GetTrajectoryStartLocation();
void UpdateTrajectoryPredictionPoints(unsigned long bForceUpdate);
void UpdateTrajectoryEnabled();
bool ShouldDrawTrajectory();
float GetAdditionalCarBounceScaleZ(class ACar_TA* Car);
void SpawnEndOfGameFX();
void SetEndOfGameHidden();
void SetExplosionFXActor(class AExplosion_X* InExplosion, class UGoal_TA* ExplosionGoal,
class AFXActor_X* ExplosionFX, class AActor* GoalOrientation);
void InitExplosionFX(class AExplosion_X* InExplosion, class UGoal_TA* ExplosionGoal, class
APRI_TA* Scorer);
void Explode(class UGoal_TA* ExplosionGoal, struct FVector ExplodeLocation, class APRI_TA*
Scorer);
void eventDestroyed();
void DoDestroy();
void DoExplode();
bool IsWorldCollision(struct FAccumulatedRigidBodyCollision& Collision);
void OnRigidBodyCollision(struct FAccumulatedRigidBodyCollision Collision);
void Launch(struct FVector LaunchPosition, struct FVector LaunchDirection);
void OnCarTouch(class ACar_TA* HitCar, uint8_t HitType);
void SetBallHitData(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal,
uint8_t HitType);
void OnHitTeamNumChanged();
bool SetHitTeamNum(uint8_t TeamNum);
void SetCarHitTeamNum(uint8_t TeamNum);
void RecordCarHit(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal,
uint8_t HitType);
void eventReset();
void OnHitWorld(struct FVector HitLoc, struct FVector HitNormal, class UPhysicalMaterial*
PhysMat);
bool IsGroundMaterial(class UPhysicalMaterial* PhysMat);
bool IsGroundHit(struct FVector HitNormal, class UPhysicalMaterial* PhysMat);
void FellOutOfWorld();
bool IsRoundActive();
void eventOnHitGoal(class UGoal_TA* Goal, struct FVector HitLoc);
void SetGameEvent(class AGameEvent_Soccar_TA* SoccarGame);
void TurnOff();
void InitAk();
void UpdateRadius();
void eventPostBeginPlay();
void eventPreBeginPlay();
void SetBallMesh(class UStaticMesh* NewMesh);
void SetWorldBounceScale(float NewScale);
void SetCarBounceScale(float NewScale);
void SetBallMaxLinearSpeedScale(float InMaxLinearSpeedScale);
void SetBallGravityScale(float InBallGravityScale);
void SetBallScale(float NewScale);
void eventReplicatedEvent(struct FName VarName);
```

```

void EventTrailIntensityChanged(class ABall_TA* Ball);
void EventHitTeamNumChanged(class ABall_TA* Ball);
void EventExploded(class ABall_TA* Ball);
void EventGameEventSet(class ABall_TA* Ball);
void EventHitGoal(class ABall_TA* Ball, class UGoal_TA* Goal);
void EventHitGround(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNormal);
void EventHitWorld(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNormal);
void EventCarTouch(class ABall_TA* Ball, class ACar_TA* HitCar, uint8_t HitType);
};

// Class TAGame.BallTrajectoryComponent_TA
// 0x006C (0x00A4 - 0x0110)
class UBallTrajectoryComponent_TA : public UActorComponent_X
{
public:
    unsigned long          bEnabled : 1;           // 0x00A8 (0x0004)
    [0x0000000000000000] [0x00000001] (CPF_Transient)
    unsigned long          bCalculatePoints : 1;   // 0x00A8 (0x0004)
    [0x0000000000000000] [0x00000002] (CPF_Transient)
    TArray<struct FVector> TrajectoryPoints;      // 0x00B0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FTrajectorySetup     TrajectorySettings; // 0x00C0 (0x0014)
    [0x0000000000000000]
    float                  TotalPathDistance;    // 0x00D4 (0x0004)
    [0x0000000000000000] (CPF_Transient)
    struct FVector          StartLocation;       // 0x00D8 (0x000C)
    [0x0000000000000000] (CPF_Transient)
    struct FVector          StartVelocity;        // 0x00E4 (0x000C)
    [0x0000000000000000] (CPF_Transient)
    float                  TrajectoryUpdateTime; // 0x00F0 (0x0004)
    [0x0000000000000000] (CPF_Transient)
    struct FScriptDelegate   __EventTrajectoryEnabledChanged__Delegate; // 0x00F8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BallTrajectoryComponent_TA");
        }

        return uClassPointer;
    }

    int32_t eventGetTrajectoryFXCount();
    void DestroyFXs();
    void eventSetupFXs();
    class APlayerController_TA* eventGetPlayer(int32_t LocalPlayerNum);
    TArray<class APlayerController_TA*> GetViewers();
    void CheckShouldShowTrajectory();
    float GetTrajectoryScale();
}

```

```

void eventUpdateTrajectoryPrediction(unsigned long bForceUpdate);
void OnEnabledChanged();
void SetEnable(unsigned long bEnable);
void HandleGameStateChanged(class AGameEvent_TA* InGameEvent);
void OnGameEventSet(class ABall_TA* Ball);
void HandleLocalPlayersChanged(class AGameEvent_TA* InGameEvent);
void eventDetached();
void eventBeginPlay();
void EventTrajectoryEnabledChanged(class ABall_TA* Ball);
};

// Class TAGame.BallTrajectoryComponent_Line_TA
// 0x0008 (0x0110 - 0x0118)
class UBallTrajectoryComponent_Line_TA : public UBallTrajectoryComponent_TA
{
public:
    unsigned long          bTrajectoryPointsNeedUpdated : 1;           // 0x0110
    (0x0004) [0x0000000000002000] [0x00000001] (CPF_Transient)
    struct FColor           LineColor;                                // 0x0114 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BallTrajectoryComponent_Line_TA");
        }

        return uClassPointer;
    };

    void DestroyFX(int32_t Index);
    void CreateFX(int32_t Index, TArray<class APlayerController_TA*>& Viewers);
    void DestroyFXs();
    void eventSetupFXs();
};

// Class TAGame.BallTrajectoryComponent_Beam_TA
// 0x0018 (0x0118 - 0x0130)
class UBallTrajectoryComponent_Beam_TA : public UBallTrajectoryComponent_Line_TA
{
public:
    TArray<struct FTrajectoryFX>      TrajectoryFXs;                // 0x0118 (0x0010)
    [0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
    class UParticleSystem*             ParticleBeamArchetype;         // 0x0128
    (0x0008) [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
};

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallTrajectoryComponent_Beam_TA");
}

return uClassPointer;
};

int32_t eventGetTrajectoryFXCount();
void DestroyFX(int32_t Index);
void DestroyFXs();
void CreateFX(int32_t Index, TArray<class APlayerController_TA*>& Viewers);
};

// Class TAGame.BallTrajectoryComponent_Spline_TA
// 0x0024 (0x0118 - 0x013C)
class UBallTrajectoryComponent_Spline_TA : public UBallTrajectoryComponent_Line_TA
{
public:
TArray<struct FTrajectorySplineFX> TrajectoryFXs; // 0x0118 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class ASplineLoftActorSpawnable_TA* SplineArchetype; // 0x0128
(0x0008) [0x0000000000000000]
float TangentMultiplier; // 0x0130 (0x0004)
[0x0000000000000000]
float PulseMultiplier; // 0x0134 (0x0004)
float MinInterpSpeed; // 0x0138 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallTrajectoryComponent_Spline_TA");
}

return uClassPointer;
};

int32_t eventGetTrajectoryFXCount();
void DestroyFX(int32_t Index);
void DestroyFXs();
void CreateFX(int32_t Index, TArray<class APlayerController_TA*>& Viewers);
void SetupFXs();
};

// Class TAGame.BallTrajectoryComponent_Mesh_TA
// 0x0040 (0x0110 - 0x0150)
class UBallTrajectoryComponent_Mesh_TA : public UBallTrajectoryComponent_TA

```

```

{
public:
TArray<struct FMeshInfo>           InterpMeshes;          // 0x0110 (0x0010)
[0x0000000000480000] (CPF_Component | CPF_NeedCtorLink)
class UMaterialInstanceConstant*      MIC;                // 0x0120 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UStaticMesh*                  StaticMesh;         // 0x0128 (0x0008)
[0x0000000000000000]
struct FColor                      BeamColor_Max;       // 0x0130 (0x0004)
[0x0000000000000000]
struct FColor                      BeamColor_Mid;        // 0x0134 (0x0004)
[0x0000000000000000]
struct FColor                      BeamColor_Min;       // 0x0138 (0x0004)
float                             MinInterpSpeed;     // 0x013C (0x0004)
[0x0000000000000000]
float                             MeshScale;          // 0x0140 (0x0004)
[0x0000000000000000]
float                             MeshMoveSpeed;      // 0x0144 (0x0004)
[0x0000000000000000]
float                             MeshRotateSpeed;    // 0x0148 (0x0004)
[0x0000000000000000]
float                             TrajectoryPlayTestScale; // 0x014C (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallTrajectoryComponent_Mesh_TA");
}

return uClassPointer;
};

struct FLinearColor GetBeamColor(float VelocityPower);
void HandleVelocityStartSpeedChanged(class ABall_GameEditor_TA* Ball);
void SetupBallEditorMaterial();
float GetTrajectoryScale();
float eventGetMeshScale(int32_t MeshIndex);
int32_t eventGetTrajectoryFXCount();
void DestroyFXs();
void eventSetupFXs();
};

// Class TAGame.BoostMesh_TA
// 0x0024 (0x0060 - 0x0084)
class UBoostMesh_TA : public UObject
{
public:
struct FRandomRange               FadeInTime;         // 0x0060 (0x0008)

```

```

[0x0000000000000001] (CPF_Edit)
struct FRandomRange           FadeOutTime;          // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FAnimatedMaterialMesh> MaterialParams; // 0x0070
(0x0010) [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
float                         MaxMaterialTime;      // 0x0080 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BoostMesh_TA");
}

return uClassPointer;
};

void ApplyPaint(class UProductAttribute_PaintSettings_TA* PaintSettings, class
UProductPaint_TA* Paint);
void ApplyLinearColorParameter(class UActorComponent* Component, struct FName Param,
struct FLinearColor InColor);
void ApplyColorParameter(class UActorComponent* Component, struct FName Param, struct
FVector Value, float Alpha);
void ApplyVectorParameter(class UActorComponent* Component, struct FName Param, struct
FVector Value);
void ApplyFloatParameter(class UActorComponent* Component, struct FName Param, float
Value);
void ApplyMaterialOverride(class UMaterialInterface* Override, int32_t Index);
void FadeOut();
void FadeIn();
void ResetElapsedTimer(float Value);
void CopyBoostMeshDistributionParameters(class AFXActor_Boost_TA* Archetype);
void Update(float DeltaTime);
void Initialize(class AFXActor_X* FXActor);
};

// Class TAGame.BreakOutActor_Platform_TA
// 0x00E8 (0x0268 - 0x0350)
class ABreakOutActor_Platform_TA : public AActor
{
public:
unsigned long             bHalfSize : 1;          // 0x0268 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long             bPrimaryPlayerStart : 1; // 0x0268 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
int32_t                  TeamIndex;            // 0x026C (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t                  MaxDamage;            // 0x0270 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UStaticMeshComponent* StaticMeshComponent; // 0x0278

```

```

(0x0008) [0x0000000040A000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF>EditInline)
class UDynamicLightEnvironmentComponent*      LightEnvironment;           //
0x0280 (0x0008) [0x0000000040A000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF>EditConst | CPF_Component | CPF>EditInline)
class AFXActor_X*                          FFArchetype;                // 0x0288 (0x0008)
[0x0000000000000001] (CPF>Edit)
float                           NeighborRadius;               // 0x0290 (0x0004)
[0x0000000000000001] (CPF>Edit)
TArray<class ABreakOutActor_Platform_TA*>   Neighbors;                 // 0x0298
(0x0010) [0x0000000000420001] (CPF>Edit | CPF>EditConst | CPF>NeedCtorLink)
class UStaticMesh*                      ConnectedCollisionMesh;        // 0x02A8 (0x0008)
[0x0000000000000001] (CPF>Edit)
class UStaticMesh*                      ConnectedCollisionMesh_HalfSize; // 0x02B0
(0x0008) [0x0000000000000001] (CPF>Edit)
class UStaticMesh*                      DisconnectedCollisionMesh;     // 0x02B8
(0x0008) [0x0000000000000001] (CPF>Edit)
class UStaticMesh*                      DisconnectedCollisionMesh_HalfSize; // 0x02C0
(0x0008) [0x0000000000000001] (CPF>Edit)
TArray<class UStaticMeshComponent*>    CollisionComponents;          // 0x02C8
(0x0010) [0x0000000004482008] (CPF_ExportObject | CPF>Transient | CPF_Component | CPF>NeedCtorLink | CPF>EditInline)
float                           LastHitTime;                  // 0x02D8 (0x0004)
[0x0000000000000000]
struct FBreakoutDamageState          DamageState;                // 0x02E0 (0x0020)
[0x0000000100000020] (CPF>Net)
class AFXActor_X*                  FFActor;                   // 0x0300 (0x0008)
[0x0000000000000000]
struct FBreakoutDamageState         PreReplayState;            // 0x0308 (0x0020)
[0x0000000000002000] (CPF>Transient)
struct FVector                     HalfSizeFirstTileScale; // 0x0328 (0x000C)
[0x0000000000002000] (CPF>Const | CPF>EditConst)
struct FScriptDelegate             __EventDamageStateChanged__Delegate; // 0x0338
(0x0018) [0x000000000400000] (CPF>NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BreakOutActor_Platform_TA");
}

return uClassPointer;
};

void SetPreReplayState();
void SavePreReplayState();
static void GatherNeighbors(class ABreakOutActor_Platform_TA* Start, int32_t Iterations,
TArray<class ABreakOutActor_Platform_TA*>& ToDamage);
void HandleTeamColorsChanged(class ATeam_TA* Team);
void HandleTeamRemoved(class UObject* TeamObj);

```

```

void HandleTeamAdded(class UObject* TeamObj);
bool IsBroken();
bool IsDamaged();
void Reset();
void SetDamageState(uint8_t InDamageState, class APRI_TA* InDamageCauser, struct FVector InLocation, unsigned long bDirectDamage, unsigned long blmmediate);
void IncreaseDamage(class APRI_TA* DamageCauser, struct FVector FromLocation, unsigned long bDirectDamage);
bool NeighborDamage(class ABall_TA* Ball, int32_t InDamage, struct FVector HitLocation);
bool OnBallHit(class ABall_TA* Ball, int32_t InDamage, struct FVector HitNormal, struct FVector InVelocity, struct FVector HitLocation);
void RefreshCollisionForComponent(class UStaticMeshComponent* Collision);
class ABreakOutActor_Platform_TA* GetClosestNeighborInDirection(struct FVector Dir);
void RefreshCollision();
void CreateCollisionComponents();
void RelnitPhysics();
void RelnitPhysicsDelayed();
void InitFX();
void eventPostBeginPlay();
void eventPreBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventDamageStateChanged(class ABreakOutActor_Platform_TA* Platform, struct FBreakoutDamageState InState);
};


```

```

// Class TAGame.BTComponent
// 0x0078 (0x0070 - 0x00E8)
class UBTComponent : public UComponent
{
public:
class UBTNode*           Root;           // 0x0070 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
uint8_t                  RootStatus;     // 0x0078 (0x0001)
[0x0000000000002002] (CPF_Const | CPF_Transient)
class AAIController_TA*   AI;            // 0x0080 (0x0008)
[0x0000004000002000] (CPF_Transient)
unsigned long             bPrintStatusLogs : 1; // 0x0088 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long             bDirtyStatusLogs : 1; // 0x0088 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long             bEvaluatingConditions : 1; // 0x0088 (0x0004)
[0x0000000000002002] [0x00000004] (CPF_Const | CPF_Transient)
unsigned long             bPauseNextTick : 1; // 0x0088 (0x0004)
[0x0000000000002000] [0x00000008] (CPF_Transient)
TArray<struct FBTStatusLog>      StatusLogs; // 0x0090 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
int32_t                  StatusLogsIndex; // 0x00A0 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
TArray<int32_t>           StatusLogsStack; // 0x00A8 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
class UDebugDrawer_X*     CanvasDrawer; // 0x00B8 (0x0008)
[0x0000000000002000] (CPF_Transient)
TArray<struct FBTDynamicLinkData>    DynamicLinks; // 0x00C0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

```

```
struct FScriptDelegate           _EventDynamicLinksChanged__Delegate;      // 0x00D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTComponent");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
struct FColor GetStatusColor(uint8_t Status);
class FString GetDepthSpaces(float Depth);
void SetDynamicLink(struct FName LinkName, class UBTNode* Node);
void SetRoot(class UBTNode* NewRootArchetype);
void Abort();
void Tick(float DeltaTime);
void Init(class AAIController_TA* ForAI);
void EventDynamicLinksChanged(class UBTComponent* BT);
};

// Class TAGame.BTLockCollection
// 0x0040 (0x0070 - 0x00B0)
class UBTLockCollection : public UComponent
{
public:
TArray<struct FTimedLock>          Locks;                      // 0x0070 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate           _EventLockSet__Delegate;      // 0x0080
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           _EventLockExpired__Delegate; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTLockCollection");
}

return uClassPointer;
};

void UpdateLocks();
float GetTimeRemaining(struct FName LockName);
```

```

bool IsNameLocked(struct FName LockName);
void RemoveNamedLock(struct FName LockName);
void AddNamedLock(struct FName LockName, float ForHowLong);
void EventLockExpired(struct FName LockName);
void EventLockSet(struct FName LockName);
};

// Class TAGame.BTNode
// 0x0020 (0x0060 - 0x0080)
class UBTNode : public UObject
{
public:
TArray<class UBTDecorator*>          Decorators;           // 0x0060 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
unsigned long                      bDebugPause : 1;      // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long                      bDebug : 1;          // 0x0070 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
class AAIController_TA*           AI;                  // 0x0078 (0x0008)
[0x0000000000002002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTNode");
}

return uClassPointer;
};

void SetStatusString(class FString msg);
};

// Class TAGame.BTConditionalNode
// 0x0030 (0x0080 - 0x00B0)
class UBTConditionalNode : public UBTNode
{
public:
TArray<class UBTNode*>          PreConditions;        // 0x0080 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
TArray<class UBTNode*>          Conditions;           // 0x0090 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
TArray<class UBTNode*>          SuccessConditions;   // 0x00A0 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTConditionalNode");
}

return uClassPointer;
};

};

// Class TAGame.BTAction
// 0x000C (0x00B0 - 0x00BC)
class UBTAction : public UBTConditionalNode
{
public:
float FinishTime; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AbortTime; // 0x00B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float StartTime; // 0x00B8 (0x0004)
[0x000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction");
}

return uClassPointer;
};

float GetRunningTime();
uint8_t OnExecute();
void eventOnStarted();
uint8_t eventOnStart();
void eventOnInit();
};

// Class TAGame.BTAction_Target
// 0x00A4 (0x00BC - 0x0160)
class UBTAction_Target : public UBTAction
{
public:
class UBTTTarget* Target; // 0x00C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float TargetPredictionTime; // 0x00C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FAIProxyData SelfData; // 0x00D0 (0x0048)
[0x0000000000002000] (CPF_Transient)
struct FAIProxyData TargetData; // 0x0118 (0x0048)
};

```

[0x0000000000000000] (CPF\_Transient)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.BTAction_Target");  
}  
  
return uClassPointer;  
};  
  
uint8_t eventOnTargetExecute();  
uint8_t OnExecute();  
};  
  
// Class TAGame.BTAction_MoveTo  
// 0x0068 (0x0160 - 0x01C8)  
class UBTAction_MoveTo : public UBTAction_Target  
{  
public:  
unsigned long          bReverse : 1;           // 0x0160 (0x0004)  
[0x0000000000000001] [0x00000001] (CPF_Edit)  
unsigned long          bBoost : 1;            // 0x0160 (0x0004)  
[0x0000000000000001] [0x00000002] (CPF_Edit)  
unsigned long          bAllowPointTurns : 1;    // 0x0160 (0x0004)  
[0x0000000000000001] [0x00000004] (CPF_Edit)  
unsigned long          bPointTurn : 1;         // 0x0160 (0x0004)  
[0x0000000000002000] [0x00000008] (CPF_Transient)  
unsigned long          bWasFacingDestination : 1; // 0x0160 (0x0004)  
[0x0000000000002000] [0x00000010] (CPF_Transient)  
unsigned long          bHasPath : 1;          // 0x0160 (0x0004)  
[0x0000000000002000] [0x00000020] (CPF_Transient)  
float                 MinSpeed;           // 0x0164 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
float                 MaxSpeed;           // 0x0168 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
float                 RandomMaxSpeedModifier; // 0x016C (0x0004)  
[0x0000000000000001] (CPF_Edit)  
float                 MinBoostDistance;     // 0x0170 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
float                 TeamSeparationDistance; // 0x0174 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
float                 ReachDistance;       // 0x0178 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
class UBTTTarget*      FaceLocation;        // 0x0180 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
class UBTTTarget*      FaceRotation;        // 0x0188 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
float                 MaxTurnRadius;       // 0x0190 (0x0004)  
[0x0000000000000001] (CPF_Edit)
```

```
 TArray<class UBTMoveModifier*>           Modifiers;          // 0x0198 (0x0010)
[0x0000000004480009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF_NeedCtorLink |
CPF_EditInline)
struct FVector             OldPathDestination;      // 0x01A8 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector             OldPathCarLocation;     // 0x01B4 (0x000C)
[0x0000000000002000] (CPF_Transient)
float                      CachedMaxGravitySlope;   // 0x01C0 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                      CurrentRandomMaxSpeedModifier; // 0x01C4 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_MoveTo");
}

return uClassPointer;
};

void SteerTowardsLocal(struct FVector LocalOffset);
void DriveAwayFrom(struct FVector TargetLocation, float TargetRadius, float BufferRadius);
bool SeparateTeammates();
void SlowDown();
void SetTemporaryMaxSpeed(float InSpeed);
void SlowForObstacles();
struct FVector GetAimedDestination(struct FVector Destination, struct FVector AimDir);
bool GetAimDir(struct FVector Destination, struct FVector& AimDir);
void DriveTo(struct FVector Destination);
void UpdatePath(struct FVector Destination);
void PathTo(struct FVector Destination);
void CacheGravityScale();
struct FVector GetLeadDestination();
bool HasReachedTarget();
bool HasPassedTarget();
uint8_t OnTargetExecute();
uint8_t eventOnStart();
void eventOnInit();
};

// Class TAGame.BTMoveModifier
// 0x0004 (0x0070 - 0x0074)
class UBTMoveModifier : public UComponent
{
public:
unsigned long             bEnabled : 1;          // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF>Edit)

public:
```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTMoveModifier");
}

return uClassPointer;
};

void eventAdjustDestination(struct FVector Goal, struct FVector& Destination);
};

// Class TAGame.BTCondition
// 0x000C (0x0080 - 0x008C)
class UBTCondition : public UBTNode
{
public:
float UpdateDelay; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bFlipResult : 1; // 0x0084 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bDirty : 1; // 0x0084 (0x0004)
[0x0000004000002002] [0x00000002] (CPF_Const | CPF_Transient)
unsigned long CachedResult : 1; // 0x0084 (0x0004)
[0x0000004000002002] [0x00000004] (CPF_Const | CPF_Transient)
float NextUpdateTime; // 0x0088 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTCondition");
}

return uClassPointer;
};

bool ShouldProcessTeam(uint8_t CheckTeam, int32_t ActorTeam);
bool ShouldProcessActor(uint8_t CheckTeam, class AActor* CheckActor);
void SetDirty();
bool eventGetResult();
void eventOnInit();
};

// Class TAGame.BTC_Any
// 0x000C (0x008C - 0x0098)
class UBTC_Any : public UBTCondition

```

```
{  
public:  
class UBTTarget* Targets; // 0x0090 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.BTC_Any");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.BTC_NotAny  
// 0x000C (0x008C - 0x0098)  
class UBTC_NotAny : public UBTCondition  
{  
public:  
class UBTTarget* Targets; // 0x0090 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.BTC_NotAny");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.BTC_ProxyBase  
// 0x005C (0x008C - 0x00E8)  
class UBTC_ProxyBase : public UBTCondition  
{  
public:  
class UBTTarget* Self; // 0x0090 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
float SelfPredictionTime; // 0x0098 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
struct FAIProxyData SelfData; // 0x00A0 (0x0048)  
[0x0000000000002000] (CPF_Transient)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_ProxyBase");
}

return uClassPointer;
};

};

// Class TAGame.BTC_TargetBase
// 0x00E8 (0x00E8 - 0x01D0)
class UBTC_TargetBase : public UBTC_ProxyBase
{
public:
class UBTTTarget* Target; // 0x00E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float TargetPredictionTime; // 0x00F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bFlattenToDrivePlane : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
struct FAIProxyData TargetData; // 0x00F8 (0x0048)
[0x0000000000002000] (CPF_Transient)
struct FAIProxyData SelfData3D; // 0x0140 (0x0048)
[0x0000000000002000] (CPF_Transient)
struct FAIProxyData TargetData3D; // 0x0188 (0x0048)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_TargetBase");
}

return uClassPointer;
};

};

// Class TAGame.BTC_Target
// 0x0048 (0x01D0 - 0x0218)
class UBTC_Target : public UBTC_TargetBase
{
public:

```

```

float DistanceGreaterThan; // 0x01D0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DistanceLessThan; // 0x01D4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float HeightGreaterThan; // 0x01D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float HeightLessThan; // 0x01DC (0x0004)
[0x0000000000000001] (CPF_Edit)
float SpeedGreaterThan; // 0x01E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SpeedLessThan; // 0x01E4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SpeedTowardsGreaterThan; // 0x01E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SpeedTowardsLessThan; // 0x01EC (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideSpeedGreaterThan; // 0x01F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideSpeedLessThan; // 0x01F4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RelativeSpeedGreaterThan; // 0x01F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RelativeSpeedLessThan; // 0x01FC (0x0004)
[0x0000000000000001] (CPF_Edit)
float RelativeSpeedTowardsGreaterThan; // 0x0200 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RelativeSpeedTowardsLessThan; // 0x0204 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RelativeSideSpeedGreaterThan; // 0x0208 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RelativeSideSpeedLessThan; // 0x020C (0x0004)
[0x0000000000000001] (CPF_Edit)
float AngleGreaterThan; // 0x0210 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AngleLessThan; // 0x0214 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Target");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_SelectActor

```

```
// 0x000C (0x008C - 0x0098)
class UBTC_SelectActor : public UBTCondition
{
public:
class UBTTarget* Targets; // 0x0090 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_SelectActor");
}

return uClassPointer;
};

};

// Class TAGame.BTAction_Input
// 0x0024 (0x00BC - 0x00E0)
class UBTAction_Input : public UBTAction
{
public:
struct FVehicleInputs Input; // 0x00C0 (0x0020)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_Input");
}

return uClassPointer;
};

uint8_t OnExecute();
};

// Class TAGame.BTAction_Stop
// 0x0008 (0x00BC - 0x00C4)
class UBTAction_Stop : public UBTAction
{
public:
unsigned long bWasGoingForward : 1; // 0x00C0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_Stop");
}

return uClassPointer;
};

uint8_t OnExecute();
uint8_t eventOnStart();
};

// Class TAGame.BTAction_HitBall
// 0x0020 (0x01C8 - 0x01E8)
class UBTAction_HitBall : public UBTAction_MoveTo
{
public:
float RandomTargetOffset; // 0x01C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AimOffsetScale; // 0x01CC (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector CurrentRandomTargetOffset; // 0x01D0 (0x000C)
[0x0000000000002000] (CPF_Transient)
class AAIController_Soccar_TA* SoccarAI; // 0x01E0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_HitBall");
}

return uClassPointer;
};

struct FVector GetAimedDestination(struct FVector Destination, struct FVector AimDir);
struct FVector GetLeadDestination();
uint8_t OnTargetExecute();
bool HasHitBall();
void OnStarted();
void OnInit();
};

// Class TAGame.BTAction_CatchBall
// 0x0004 (0x01E8 - 0x01EC)

```

```
class UBTAction_CatchBall : public UBTAction_HitBall
{
public:
float FallTime; // 0x01E8 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_CatchBall");
}

return uClassPointer;
};

void DriveTo(struct FVector Destination);
uint8_t OnTargetExecute();
};

// Class TAGame.BTAction_WiggleTires
// 0x0004 (0x00BC - 0x00C0)
class UBTAction_WiggleTires : public UBTAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_WiggleTires");
}

return uClassPointer;
};

uint8_t OnExecute();
};

// Class TAGame.BTNodeParent
// 0x0014 (0x00B0 - 0x00C4)
class UBTNodeParent : public UBTConditionalNode
{
public:
TArray<class UBTNode*> Children; // 0x00B0 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
int32_t RunningChild; // 0x00C0 (0x0004)
[0x0000000000000002] (CPF_Const | CPF_Transient)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTNodeParent");
}

return uClassPointer;
};

};

// Class TAGame.BTExecutor
// 0x0004 (0x00C4 - 0x00C8)
class UBTExecutor : public UBTNodeParent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTExecutor");
}

return uClassPointer;
};

};

// Class TAGame.BTRandomSelector
// 0x0024 (0x00C4 - 0x00E8)
class UBTRandomSelector : public UBTNodeParent
{
public:
TArray<float> ChildWeights; // 0x00C8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t> RemainingChildren; // 0x00D8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```
uClassPointer = UObject::FindClass("Class TAGame.BTRandomSelector");
}

return uClassPointer;
};

};

// Class TAGame.BTSelector
// 0x0004 (0x00C4 - 0x00C8)
class UBTSelector : public UBTNodeParent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTSelector");
}

return uClassPointer;
};

};

// Class TAGame.BTEvaluator
// 0x0000 (0x00C8 - 0x00C8)
class UBTEvaluator : public UBTSelector
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTEvaluator");
}

return uClassPointer;
};

};

// Class TAGame.BTSequence
// 0x0004 (0x00C4 - 0x00C8)
class UBTSequence : public UBTNodeParent
{
```

```
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTSequence");
}

return uClassPointer;
};

};

// Class TAGame.BTDecorator
// 0x0008 (0x0080 - 0x0088)
class UBTDecorator : public UBTNode
{
public:
class UBTNode*           Child;           // 0x0080 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTDecorator");
}

return uClassPointer;
};

};

// Class TAGame.BTDeco_RandomSteer
// 0x0010 (0x0088 - 0x0098)
class UBTDeco_RandomSteer : public UBTDecorator
{
public:
float                  MinDuration;      // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                  MaxDuration;      // 0x008C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                  TimeRemaining;    // 0x0090 (0x0004)
[0x0000000000000000]
float                  ExtraSteer;       // 0x0094 (0x0004)
[0x0000000000000000]
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTDeco_RandomSteer");
}

return uClassPointer;
};

void eventAdjustSteer();
};

// Class TAGame.BTDeco_Success
// 0x0000 (0x0088 - 0x0088)
class UBTDeco_Success : public UBTDecorator
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTDeco_Success");
}

return uClassPointer;
};

};

// Class TAGame.BTDynamicLink
// 0x0018 (0x0088 - 0x00A0)
class UBDynamicLink : public UBTDecorator
{
public:
struct FName           LinkName;          // 0x0088 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
unsigned long          bUnlinkedSuccess : 1; // 0x0090 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class UBTNode*         ChildArchetype;    // 0x0098 (0x0008)
[0x0000000000002002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTDynamicLink");
}

return uClassPointer;
};

void SetChildArchetype(class UBTNode* NewChildArchetype);
void HandleDynamicLinksChanged(class UBTComponent* BT);
void eventInit();
};

// Class TAGame.BTLock
// 0x0024 (0x0088 - 0x00AC)
class UBTLock : public UBTDecorator
{
public:
uint8_t LockScope; // 0x0088 (0x0001)
[0x0000000000000001] (CPF_Edit)
struct FName LockName; // 0x008C (0x0008)
[0x0000000000000001] (CPF_Edit)
float Time; // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RandomTime; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UBTLockCollection* Locks; // 0x00A0 (0x0008)
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float LockTime; // 0x00A8 (0x0004)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTLock");
}

return uClassPointer;
};

};

// Class TAGame.BTWeight
// 0x0004 (0x0088 - 0x008C)
class UBTWeight : public UBTDecorator
{
public:
float Weight; // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTWeight");
}

return uClassPointer;
};

};

// Class TAGame.BTProxyType
// 0x0004 (0x0060 - 0x0064)
class UBTProxyType : public UObject
{
public:
unsigned long           bStatic : 1;                                // 0x0060 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTProxyType");
}

return uClassPointer;
};

};

// Class TAGame.BTTTarget
// 0x0030 (0x0060 - 0x0090)
class UBTTarget : public UObject
{
public:
class UBTProxyType*          Type;                                // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector               LocalOffset;                         // 0x0068 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector               WorldOffset;                         // 0x0074 (0x000C)
[0x0000000000000001] (CPF_Edit)
TArray<class UBTNode*>      Filters;                            // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTTTarget");
}

return uClassPointer;
};

};

// Class TAGame.CachedUnlockedProducts_TA
// 0x0028 (0x0060 - 0x0088)
class UCachedUnlockedProducts_TA : public UObject
{
public:
TArray<int32_t> Products; // 0x0060 (0x0010)
[0x0008004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __Products_ChangeNotify; // 0x0070
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CachedUnlockedProducts_TA");
}

return uClassPointer;
};

void __Products_ChangeNotifyFunc();
int32_t Length();
void Clear();
void Set(TArray<int32_t>& ProductIDs);
void Remove(int32_t ProductID);
void Add(int32_t ProductID);
TArray<class UProduct_TA*> GetProducts();
bool ContainsAllProducts(class UProductSlot_TA* Slot, uint8_t UnlockMethod);
bool ContainsProductsForSlot(class UProductSlot_TA* Slot);
bool Contains(int32_t ProductID);
};

// Class TAGame.CarColorSet_TA
// 0x0054 (0x0080 - 0x00D4)
class UCarColorSet_TA : public UColorPalette_X
{
public:
TArray<struct FLinearColor> Grayscales; // 0x0080 (0x0010)

```

```

[0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FLinearColor> H; // 0x0090 (0x0010)
[0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FLinearColor> SV; // 0x00A0 (0x0010)
[0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
struct FLinearColor ColorBlindColor; // 0x00B0 (0x0010)
[0x0000000000000001] (CPF_Edit)
TArray<struct FDebugColor> DebugColors; // 0x00C0 (0x0010)
[0x0000000800400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
float MinStadiumBrightness; // 0x00D0 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarColorSet_TA");
}

return uClassPointer;
};

void RebuildColors();
struct FLinearColor GetFontColor(int32_t ColorID);
float GetMinStadiumBrightness();
TArray<struct FLinearColor> eventGetStadiumColors(int32_t ColorID);
};

// Class TAGame.CarComponent_TA
// 0x0078 (0x0268 - 0x02E0)
class ACarComponent_TA : public AActor
{
public:
class AFXActor_X* FXActorArchetype; // 0x0268 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UFXActorEvent_X* FXEvent; // 0x0270 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long bDisabled : 1; // 0x0278 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long bAutoActivate : 1; // 0x0278 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bSimulateComponent : 1; // 0x0278 (0x0004)
[0x0000000000000003] [0x00000004] (CPF_Edit | CPF_Const)
unsigned long bCreated : 1; // 0x0278 (0x0004)
[0x0000004000002000] [0x00000008] (CPF_Transient)
unsigned long bActive : 1; // 0x0278 (0x0004)
[0x0000004000002000] [0x00000010] (CPF_Transient)
unsigned long bRemovedFromCar : 1; // 0x0278 (0x0004)
[0x0000004000002000] [0x00000020] (CPF_Transient)
uint8_t ComponentData; // 0x027C (0x0001)
[0x0000000000000002] (CPF_Transient)

```

```

uint8_t ReplicatedActive; // 0x027D (0x0001)
[0x0000004100002020] (CPF_Net | CPF_Transient)
class APRI_TA* Activator; // 0x0280 (0x0008)
[0x0000000000002000] (CPF_Transient)
class AVehicle_TA* Vehicle; // 0x0288 (0x0008)
[0x000000410000020] (CPF_Net)
class ACar_TA* Car; // 0x0290 (0x0008)
[0x0000004000002000] (CPF_Transient)
float ActivityTime; // 0x0298 (0x0004)
[0x0000000000002000] (CPF_Transient)
float MinInactiveTime; // 0x029C (0x0004)
[0x0000000000000002] (CPF_Const)
float ReplicatedActivityTime; // 0x02A0 (0x0004)
[0x0000004100002020] (CPF_Net | CPF_Transient)
class AFXActor_X* FXActor; // 0x02A8 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FScriptDelegate __EventPreActivationChanged__Delegate; // 0x02B0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPostActivationChanged__Delegate; // 0x02C8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
void eventFellOutOfWorld();
void HandleCarDestroyed(class APawn_X* InCar);
void eventDestroyed();
float GetInactiveTime();
float GetActiveTime();
void ApplyForces(float ActiveTime);
void PrePhysicsStep(float DeltaTime);
void RemoveFromCar();
bool CanDeactivate();
bool ConditionalDeactivate();
bool CanActivate();
bool ConditionalActivate();
void SetActive(unsigned long bSetActive);
void Deactivate();
void Activate();
void RemoveFX();
void RelinitFX();
void InitFX();
void UnregisterCarEvents();

```

```

void RegisterCarEvents();
void OnCreated();
void HandleVehicleSetup(class ACar_TA* InCar);
void OnVehicleSetupComplete();
void Create(class ACar_TA* OwnerCar, class APRI_TA* InActivator);
class ACarComponent_TA* ArchetypeCreateFor(class ACar_TA* OwnerCar);
void ClientUpdateActive();
void eventReplicatedEvent(struct FName VarName);
void EventPostActivationChanged(class ACarComponent_TA* CarComponent);
void EventPreActivationChanged(class ACarComponent_TA* CarComponent);
};

// Class TAGame.CarComponent_AirActivate_TA
// 0x0020 (0x02E0 - 0x0300)
class ACarComponent_AirActivate_TA : public ACarComponent_TA
{
public:
int32_t AirActivateCount; // 0x02E0 (0x0004)
[0x0000004100002020] (CPF_Net | CPF_Transient)
int32_t MaxAirActivateCount; // 0x02E4 (0x0004)
[0x0000004000000002] (CPF_Const)
struct FScriptDelegate __EventAirActivateCountChanged__Delegate; // 0x02E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_AirActivate_TA");
}

return uClassPointer;
};

void __CarComponent_AirActivate_TA__OnVehicleSetupComplete_0x1(class ACar_TA* _);
bool CanActivate();
int32_t GetAirCountRemaining();
void OnAirActivateCountChanged();
void SetAirActivateCount(int32_t Count);
void OnVehicleSetupComplete();
void eventReplicatedEvent(struct FName VarName);
void EventAirActivateCountChanged(class ACarComponent_AirActivate_TA* CarComponent);
};

// Class TAGame.CarComponent_Boost_TA
// 0x0070 (0x0300 - 0x0370)
class ACarComponent_Boost_TA : public ACarComponent_AirActivate_TA
{
public:
float BoostConsumptionRate; // 0x0300 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

float MaxBoostAmount; // 0x0304 (0x0004)
[0x0000000000000001] (CPF_Edit)
float StartBoostAmount; // 0x0308 (0x0004)
[0x0000000000000021] (CPF_Edit | CPF_Net)
float CurrentBoostAmount; // 0x030C (0x0004)
[0x000008000000021] (CPF_Edit | CPF_Net)
float BoostModifier; // 0x0310 (0x0004)
[0x0000000000000021] (CPF_Edit | CPF_Net)
float LastBoostAmountRequestTime; // 0x0314 (0x0004)
[0x0000000000002000] (CPF_Transient)
float LastBoostAmount; // 0x0318 (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long bPendingConfirmBoostAmount : 1; // 0x031C
(0x0004) [0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long bRechargeGroundOnly : 1; // 0x031C (0x0004)
[0x0000000000000020] [0x00000002] (CPF_Net)
unsigned long bNoBoost : 1; // 0x031C (0x0004)
[0x0000000100000021] [0x00000004] (CPF_Edit | CPF_Net)
float BoostForce; // 0x0320 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinBoostTime; // 0x0324 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RechargeRate; // 0x0328 (0x0004)
[0x0000000000000020] (CPF_Net)
float RechargeDelay; // 0x032C (0x0004)
[0x0000000000000020] (CPF_Net)
int32_t UnlimitedBoostRefCount; // 0x0330 (0x0004)
[0x0000000000000021] (CPF_Edit | CPF_Net)
struct FReplicatedBoostData ReplicatedBoost; // 0x0334 (0x0004)
[0x0000000100002020] (CPF_Net | CPF_Transient)
uint8_t ReplicatedBoostAmount; // 0x0338 (0x0001)
[0x0000000100002020] (CPF_Net | CPF_Transient)
class UProductAsset_Boost_TA* BoostAsset; // 0x0340 (0x0008)
[0x0000040000000000]
class UFXActorEvent_X* BoostShakeState; // 0x0348 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProfileControlsSave_TA* ControlsSave; // 0x0350 (0x0008)
[0x0000000000000000]
struct FScriptDelegate __EventBoostAmountChanged__Delegate; // 0x0358
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Boost_TA");
}

return uClassPointer;
};

```

```

bool ShouldPredictBoostConsumption();
void __CarComponent_Boost_TA__OnCreated_0x1(class UObject* Obj);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void ReadReplicatedBoostAmount();
void Deprecated_ReadReplicatedBoostAmount();
void eventSetReplicatedBoostAmount();
void ApplyForces(float ActiveTime);
void PrePhysicsStep(float DeltaTime);
bool ActivateDriveThrottle();
void ClientGiveBoost(float Amount);
void ConfirmBoostAmount();
void SendConfirmBoostAmount();
void ClientFixBoostAmount(float TimeStamp, float Amount);
void ServerConfirmBoostAmount(float TimeStamp, float Amount);
void SetRechargeGroundOnly(unsigned long bInGroundOnly);
void SetRechargeDelay(float InRechargeDelay);
void SetRechargeRate(float InRechargeRate);
bool HasUnlimitedBoost();
void SetNoBoost(unsigned long Enabled);
void SetUnlimitedBoost(unsigned long Enabled);
void SetUnlimitedBoostDelayed(unsigned long Enabled);
void SetBoostModifier(float Modifier);
void SetBoostAmount(float Amount);
void GiveBoost(float Amount);
void GiveStartingBoost();
void GiveFullBoost();
float GetPercentBoostFull();
bool IsFull();
void RemoveFromCar();
bool CanDeactivate();
bool CanActivate();
void InitFX();
void HandleBoostAsset(class UProductAsset_Boost_TA* Asset);
void OnCreated();
void eventReplicatedEvent(struct FName VarName);
void EventBoostAmountChanged(class ACarComponent_Boost_TA* Boost);
};

// Class TAGame.CarComponent_Boost_KO_TA
// 0x0008 (0x0370 - 0x0378)
class ACarComponent_Boost_KO_TA : public ACarComponent_Boost_TA
{
public:
float LimitVelocityXYToForceScale; // 0x0370 (0x0004)
[0x0001000000000002] (CPF_Const)
float LimitVelocityZToForceScale; // 0x0374 (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Boost_KO_TA");
}

return uClassPointer;
};

void ApplyForces(float ActiveTime);
};

// Class TAGame.CarComponent_Dodge_TA
// 0x0098 (0x0300 - 0x0398)
class ACarComponent_Dodge_TA : public ACarComponent_AirActivate_TA
{
public:
float DodgeInputThreshold; // 0x0300 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideDodgeImpulse; // 0x0304 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideDodgeImpulseMaxSpeedScale; // 0x0308 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ForwardDodgeImpulse; // 0x030C (0x0004)
[0x0000000000000001] (CPF_Edit)
float ForwardDodgeImpulseMaxSpeedScale; // 0x0310 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BackwardDodgeImpulse; // 0x0314 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BackwardDodgeImpulseMaxSpeedScale; // 0x0318
(0x0004) [0x0000000000000001] (CPF_Edit)
float SideDodgeTorque; // 0x031C (0x0004)
[0x0000000000000001] (CPF_Edit)
float ForwardDodgeTorque; // 0x0320 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DodgeTorqueTime; // 0x0324 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinDodgeTorqueTime; // 0x0328 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DodgeZDamping; // 0x032C (0x0004)
[0x0000000000000001] (CPF_Edit)
float DodgeZDampingDelay; // 0x0330 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DodgeZDampingUpTime; // 0x0334 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DodgeImpulseScale; // 0x0338 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DodgeTorqueScale; // 0x033C (0x0004)
[0x0000000000000001] (CPF_Edit)
class UFXActorEvent_X* JumpEvent; // 0x0340 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UFXActorEvent_X* DoubleJumpEvent; // 0x0348 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UFXActorEvent_X* DodgeEvent; // 0x0350 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector DodgeTorque; // 0x0358 (0x000C)

```

```

[0x0000004000002020] (CPF_Net | CPF_Transient)
struct FVector DodgeDirection; // 0x0364 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FVector DodgeImpulse; // 0x0370 (0x000C)
[0x0000000020002000] (CPF_Transient | CPF_Deprecated)
struct FScriptDelegate __EventActivateDodge__Delegate; // 0x0380
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Dodge_TA");
}

return uClassPointer;
};

void SetDodgeSettings();
void PrePhysicsStep(float DeltaTime);
void ApplyTorqueForces(float ActiveTime);
void ApplyDodgeImpulse();
struct FVector GetDodgeImpulse(struct FVector DodgeDir);
void ApplyForces(float ActiveTime);
bool CanActivate();
void OnCreated();
void EventActivateDodge(class ACarComponent_Dodge_TA* DodgeComponent);
};

// Class TAGame.CarComponent_Dodge_KO_TA
// 0x0024 (0x0398 - 0x03BC)
class ACarComponent_Dodge_KO_TA : public ACarComponent_Dodge_TA
{
public:
float MaxForceToVelocityDegrees; // 0x0398 (0x0004)
[0x0001000000000002] (CPF_Const)
float MaxForceToRotationDegrees; // 0x039C (0x0004)
[0x0001000000000002] (CPF_Const)
float MinVelocityTransferAlpha; // 0x03A0 (0x0004)
[0x0001000000000002] (CPF_Const)
float MaxVelocityTransferAlpha; // 0x03A4 (0x0004)
[0x0001000000000002] (CPF_Const)
float VelocityTransferBlend; // 0x03A8 (0x0004)
[0x0001000000000002] (CPF_Const)
struct FRotator DodgeRotation; // 0x03AC (0x000C)
[0x0001004000002000] (CPF_Transient)
int32_t DodgeRotationCompressed; // 0x03B8 (0x0004)
[0x0001000100002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Dodge_KO_TA");
}

return uClassPointer;
};

struct FVector GetDodgeImpulse(struct FVector DodgeDir);
void ApplyDodgeImpulse();
void SetDodgeRotationFromCompressed(int32_t Compressed);
void eventSetDodgeRotation(struct FRotator InRotation);
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.CarComponent_DoubleJump_TA
// 0x0014 (0x0300 - 0x0314)
class ACarComponent_DoubleJump_TA : public ACarComponent_AirActivate_TA
{
public:
float JumplImpulse; // 0x0300 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ImpulseScale; // 0x0304 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector DoubleJumplImpulse; // 0x0308 (0x000C)
[0x0000004000002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_DoubleJump_TA");
}

return uClassPointer;
};

void PrePhysicsStep(float DeltaTime);
void ApplyForces(float ActiveTime);
bool CanActivate();
void OnCreated();
};

// Class TAGame.CarComponent_DoubleJump_KO_TA
// 0x0008 (0x0314 - 0x031C)
class ACarComponent_DoubleJump_KO_TA : public ACarComponent_DoubleJump_TA
{
public:

```

```

float LimitVelocityToForceScale; // 0x0318 (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_DoubleJump_KO_TA");
}

return uClassPointer;
};

};

// Class TAGame.CarComponent_DoubleJump_Robin_TA
// 0x001C (0x0314 - 0x0330)
class ACarComponent_DoubleJump_Robin_TA : public ACarComponent_DoubleJump_TA
{
public:
struct FWaveParams RandomPitch; // 0x0318 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FWaveParams RandomYaw; // 0x0320 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FWaveParams RandomRoll; // 0x0328 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_DoubleJump_Robin_TA");
}

return uClassPointer;
};

void ApplyForces(float ActiveTime);
};

// Class TAGame.CarComponent_AirControl_TA
// 0x0020 (0x02E0 - 0x0300)
class ACarComponent_AirControl_TA : public ACarComponent_TA
{
public:
struct FRotator AirTorque; // 0x02E0 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator AirDamping; // 0x02EC (0x000C)

```

```

[0x0000000000000001] (CPF_Edit)
float ThrottleForce; // 0x02F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AirControlSensitivity; // 0x02FC (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_AirControl_TA");
}

return uClassPointer;
};

void PrePhysicsStep(float DeltaTime);
void ApplyForces(float ActiveTime);
void OnCreated();
};

// Class TAGame.CarComponent_AirControl_Robin_TA
// 0x0018 (0x0300 - 0x0318)
class ACarComponent_AirControl_Robin_TA : public ACarComponent_AirControl_TA
{
public:
struct FWaveParams RandomPitch; // 0x0300 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FWaveParams RandomYaw; // 0x0308 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FWaveParams RandomRoll; // 0x0310 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_AirControl_Robin_TA");
}

return uClassPointer;
};

void ApplyForces(float ActiveTime);
};

// Class TAGame.CarComponent_FlipCar_TA
// 0x0010 (0x02E0 - 0x02F0)

```

```

class ACarComponent_FlipCar_TA : public ACarComponent_TA
{
public:
    float           FlipCarImpulse;          // 0x02E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           FlipCarTorque;          // 0x02E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           FlipCarTime;            // 0x02E8 (0x0004)
    [0x0000000000000021] (CPF_Edit | CPF_Net)
    unsigned long   bFlipRight : 1;         // 0x02EC (0x0004)
    [0x0000000000002020] [0x00000001] (CPF_Net | CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CarComponent_FlipCar_TA");
        }

        return uClassPointer;
    }

    void InitFlip();
    void PrePhysicsStep(float DeltaTime);
    void ApplyForces(float ActiveTime);
    bool CanActivate();
    void OnCreated();
};

// Class TAGame.CarComponent_Jump_TA
// 0x0028 (0x02E0 - 0x0308)
class ACarComponent_Jump_TA : public ACarComponent_TA
{
public:
    float           MinJumpTime;           // 0x02E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           JumplImpulse;          // 0x02E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           JumpForce;             // 0x02E8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           JumpForceTime;         // 0x02EC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           PodiumJumpForceTime;   // 0x02F0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           JumplImpulseSpeed;     // 0x02F4 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float           JumpAccel;             // 0x02F8 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float           MaxJumpHeight;         // 0x02FC (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float           MaxJumpHeightTime;    // 0x0300 (0x0004)
};

```

```
[0x0000000000000000] (CPF_Transient)
unsigned long bDeactivate : 1; // 0x0304 (0x0004)
[0x0000000000000000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Jump_TA");
}

return uClassPointer;
};

void PrePhysicsStep(float DeltaTime);
void ApplyForces(float ActiveTime);
void CacheJumpData();
void OnCreated();
};

// Class TAGame.CarComponent_Jump_Robin_TA
// 0x0018 (0x0308 - 0x0320)
class ACarComponent_Jump_Robin_TA : public ACarComponent_Jump_TA
{
public:
struct FWaveParams [0x0000000000000001] (CPF_Edit) RandomPitch; // 0x0308 (0x0008)
struct FWaveParams [0x0000000000000001] (CPF_Edit) RandomYaw; // 0x0310 (0x0008)
struct FWaveParams [0x0000000000000001] (CPF_Edit) RandomRoll; // 0x0318 (0x0008)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Jump_Robin_TA");
}

return uClassPointer;
};

void ApplyForces(float ActiveTime);
};

// Class TAGame.CarComponent_Torque_TA
// 0x0018 (0x02E0 - 0x02F8)
class ACarComponent_Torque_TA : public ACarComponent_TA
```

```

{
public:
float TorqueScale; // 0x02E0 (0x0004)
[0x0001000000002020] (CPF_Net | CPF_Transient)
struct FVector TorqueInput; // 0x02E4 (0x000C)
[0x0001000000002000] (CPF_Transient)
int32_t ReplicatedTorqueInput; // 0x02F0 (0x0004)
[0x0001000100002020] (CPF_Net | CPF_Transient)
float TorqueTimeScale; // 0x02F4 (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarComponent_Torque_TA");
}

return uClassPointer;
};

float eventGetTimeAlpha();
void SetTorqueInputFromCompressed(int32_t Compressed);
void Start(float Scale, struct FVector WorldDirection);
void ApplyForces(float ActiveTime);
void OnCreated();
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.SpecialPickup_TA
// 0x00A8 (0x02E0 - 0x0388)
class ASpecialPickup_TA : public ACarComponent_TA
{
public:
class FString PickupName; // 0x02E0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bHudIgnoreUseTime : 1; // 0x02F0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bHasActivated : 1; // 0x02F0 (0x0004)
[0x0000040000000000] [0x00000002]
unsigned long bIsActive : 1; // 0x02F0 (0x0004)
[0x0000008000002000] [0x00000004] (CPF_Transient)
class UTexture2D* ActiveIcon; // 0x02F8 (0x0008)
[0x0000000000000000]
class UTexture2D* InactiveIcon; // 0x0300 (0x0008)
[0x0000000000000000]
float ActivationDuration; // 0x0308 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X* PickupFXArchetype; // 0x0310 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FPickupReplacement> BodyReplacements; // 0x0318
}

```

```

(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UShakeComponent_X*           ShakeComponentArchetype;          // 0x0328
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF_EditInline)
class UAkSoundCue*               ActivateSFX;                  // 0x0330 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*               LocalActivateSFX;             // 0x0338 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X*                PickupFX;                   // 0x0340 (0x0008)
[0x0000000000000000]
class UShakeComponent_X*          ShakeComp;                  // 0x0348 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
int32_t                          PickupSlot;                 // 0x0350 (0x0004)
[0x0001000000000000]
int32_t                          CooldownSeconds;           // 0x0354 (0x0004)
[0x0001000000000020] (CPF_Net)
struct FScriptDelegate            __EventActivated__Delegate; // 0x0358
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate            __EventDeactivated__Delegate; // 0x0370
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_TA");
}

return uClassPointer;
};

bool DisableBallImpactForces();
bool GetShouldHideActivateUI();
bool HasActivated();
class ARBActor_TA* GetClientTarget();
void OnVehicleSetupComplete();
void RemovePickupFX();
void InitPickupFX();
float GetActiveTimePercent();
void PreActivatedTick(float DeltaTime);
void PickupTick(float DeltaTime);
void PickupEnd();
void PickupStart();
class ACarComponent_Boost_TA* GetBoostComponent();
void UsePickupReleased();
void DeactivatePickup();
bool TryActivate(class ARBActor_TA* TargetOverride);
void eventDestroyed();
void HandleDemolish(class ACar_TA* InCar, struct FDemolishData Data);
void OnCreated();
bool CanPickup(class ACar_TA* InCar);

```

```

class ASpecialPickup_TA* ApplyPickup(class ACar_TA* InCar);
void eventRecordHit(class ARBActor_TA* HitActor, struct FVector Normal, uint8_t HitType);
void EventDeactivated(class ACar_TA* InCar, class ASpecialPickup_TA* Pickup);
void EventActivated(class ACar_TA* InCar, class ASpecialPickup_TA* Pickup);
};

// Class TAGame.SpecialPickup_BallGravity_TA
// 0x0098 (0x0388 - 0x0420)
class ASpecialPickup_BallGravity_TA : public ASpecialPickup_TA
{
public:
    float BallGravity; // 0x0388 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float Range; // 0x038C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FVector Offset; // 0x0390 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    unsigned long bDeactivateOnTouch : 1; // 0x039C (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    class UParticleSystem* BeamPSArchetype; // 0x03A0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    struct FName BeamPSPParam; // 0x03A8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float RecordBallHitRate; // 0x03B0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UMaterialInstanceConstant* BallIMIC; // 0x03B8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    struct FName BallIMICParam; // 0x03C0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class AFXActor_TA* BallIFXArchetype; // 0x03C8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    struct FInterpCurveFloat BallIMICCurve; // 0x03D0 (0x0018)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class UAkSoundCue* BallSFX; // 0x03E8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float LastRecordedBallHitTime; // 0x03F0 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    class UParticleSystemComponent* BeamPSC; // 0x03F8
    (0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    class AFXActor_TA* BallIFX; // 0x0400 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class UStaticMeshComponent* CopiedBallMesh; // 0x0408
    (0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    class UMaterialInstanceConstant* BMIC; // 0x0410 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class ABall_TA* PrevBall; // 0x0418 (0x0008)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BallGravity_TA");
}

return uClassPointer;
};

void HandleHitBall(class ACar_TA* InCar, class ABall_TA* Ball, struct FVector HitLocation, struct
FVector HitNormal);
void RemoveBallFX();
void UpdateBallFX(class ABall_TA* Ball, struct FVector BallContactPos);
void RemoveBeamFX();
void UpdateBeamFX(class ABall_TA* Ball);
class ABall_TA* GetBallInRange();
void UpdateVisual();
void PickupTick(float DeltaTime);
void ApplyForces(float ActiveTime);
void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_HauntedBallBeam_TA
// 0x0040 (0x0420 - 0x0460)
class ASpecialPickup_HauntedBallBeam_TA : public ASpecialPickup_BallGravity_TA
{
public:
unsigned long           bIsPhasingBall : 1;           // 0x0420 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long           bIsInRange : 1;           // 0x0420 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long           bSfxLoopIsPlaying : 1;       // 0x0420 (0x0004)
[0x0000000000000000] [0x00000004]
class ABall_Haunted_TA*          PrevHauntedBall;      // 0x0428 (0x0008)
[0x0000000000000000]
int32_t                 MaxSimulatedClients;        // 0x0430 (0x0004)
[0x0000000000000000]
float                   ArrivalDistance;           // 0x0434 (0x0004)
[0x0000000000000000]
struct FVector           CarAvoidenceOffset;        // 0x0438 (0x000C)
[0x0000000000000000]
class UAkSoundCue*        BeamEndSFX;           // 0x0448 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*        ConnectedBeamLoopSFX;    // 0x0450
(0x0008) [0x0000000000000001] (CPF_Edit)
class UAkSoundCue*        BeamLoopSFX;           // 0x0458 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_HauntedBallBeam_TA");
}

return uClassPointer;
};

void RemoveBeamFX();
void UpdateBeamFX(class ABall_TA* Ball);
void RemovePhase();
class ABall_TA* GetBallInRange();
void PickupTick(float DeltaTime);
void PickupEnd();
void PickupStart();
void ApplyForces(float ActiveTime);
};

// Class TAGame.SpecialPickup_Targeted_TA
// 0x0038 (0x0388 - 0x03C0)
class ASpecialPickup_Targeted_TA : public ASpecialPickup_TA
{
public:
unsigned long          bCanTargetBall : 1;           // 0x0388 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bCanTargetCars : 1;           // 0x0388 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long          bCanTargetEnemyCars : 1;       // 0x0388 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long          bCanTargetTeamCars : 1;         // 0x0388 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
unsigned long          bUseDirectionalTargeting : 1;    // 0x0388 (0x0004)
[0x0000000000000001] [0x00000010] (CPF_Edit)
unsigned long          bRequireTrace : 1;             // 0x0388 (0x0004)
[0x0000000000000001] [0x00000020] (CPF_Edit)
float                 Range;                          // 0x038C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                 DirectionalTargetingAccuracy; // 0x0390 (0x0004)
[0x0000000000000001] (CPF_Edit)
class ARBActor_TA*     ClientTarget;            // 0x0398 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ARBActor_TA*     Targeted;                // 0x03A0 (0x0008)
[0x0000000100002020] (CPF_Net | CPF_Transient)
struct FScriptDelegate _EventNewTarget__Delegate; // 0x03A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Targeted_TA");
}
}

```

```

}

return uClassPointer;
};

class ARBActor_TA* GetClientTarget();
void TargetChanged();
void OnTargetChanged();
void PreActivatedTick(float DeltaTime);
bool TryActivate(class ARBActor_TA* TargetOverride);
bool ValidateTargetTrace(class ARBActor_TA* InTarget);
bool ValidateTarget(class ARBActor_TA* InTarget);
class ARBActor_TA* GetTarget();
void eventReplicatedEvent(struct FName VarName);
void EventNewTarget();
};

// Class TAGame.SpecialPickup_BallFreeze_TA
// 0x0098 (0x03C0 - 0x0458)
class ASpecialPickup_BallFreeze_TA : public ASpecialPickup_Targeted_TA
{
public:
class AFXActor_X* [0x0000000000000001] (CPF_Edit) FreezeBreakFXArchetype; // 0x03C0 (0x0008)
class AFXActor_X* [0x0000000000000001] (CPF_Edit) FreezeFXArchetype; // 0x03C8 (0x0008)
class UMaterialInstanceConstant* [0x0000000000000001] (CPF_Edit) FreezeMIC; // 0x03D0 (0x0008)
struct FName [0x0000000000000001] (CPF_Edit) FreezeMaterialParam; // 0x03D8 (0x0008)
struct FInterpCurveFloat [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) FreezelInterpTime; // 0x03E0 (0x0018)
unsigned long [0x0000000000000001] [0x00000001] (CPF_Edit) bMaintainMomentum : 1; // 0x03F8 (0x0004)
unsigned long [0x0000000000000001] [0x00000002] (CPF_Transient) bTouched : 1; // 0x03F8 (0x0004)
float [0x0000000000000001] (CPF_Edit) TimeToStop; // 0x03FC (0x0004)
float [0x0000000000000001] (CPF_Edit) StopMomentumPercentage; // 0x0400 (0x0004)
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) FreezeSFX; // 0x0408 (0x0008)
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) BreakSFX; // 0x0410 (0x0008)
class UStaticMeshComponent* (0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline) CopiedBallMesh; // 0x0418
class ABall_TA* [0x0000000000000001] (CPF_Transient) Ball; // 0x0420 (0x0008)
struct FVector [0x0000000000000001] (CPF_Transient) OrigLinearVelocity; // 0x0428 (0x000C)
struct FVector [0x0000000000000001] (CPF_Transient) OrigAngularVelocity; // 0x0434 (0x000C)
float [0x0000000000000001] (CPF_Transient) OrigSpeed; // 0x0440 (0x0004)

```

```

[0x0000000000000000] (CPF_Transient)
float RepOrigSpeed; // 0x0444 (0x0004)
[0x0000000000000000] (CPF_Net | CPF_Transient)
class UMaterialInstanceConstant* FreezeMat; // 0x0448 (0x0008)
[0x0000000000000000] (CPF_Transient)
class AFXActor_X* FreezeFX; // 0x0450 (0x0008)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BallFreeze_TA");
}

return uClassPointer;
};

void PickupEnd();
void SpawnBreakFX();
void HandleBallExploded(class ABall_TA* InBall);
void HandleBallHit(class ABall_TA* InBall, class ACar_TA* InCar, uint8_t HitType);
void RemovePickupFX();
void TickPickupFX(float DeltaTime);
void ApplyForces(float ActiveTime);
void PickupTick(float DeltaTime);
void InitPickupFX();
void OnTargetChanged();
void PickupStart();
};

// Class TAGame.SpecialPickup_GrapplingHook_TA
// 0x0108 (0x03C0 - 0x04C8)
class ASpecialPickup_GrapplingHook_TA : public ASpecialPickup_Targeted_TA
{
public:
float Impulse; // 0x03C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float Force; // 0x03C4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxRopeLength; // 0x03C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float PredictionSpeed; // 0x03CC (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bDeactivateOnTouch : 1; // 0x03D0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bInstant : 1; // 0x03D0 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bBlocked : 1; // 0x03D0 (0x0004)
[0x0000000000000000] [0x00000004] (CPF_Transient)
unsigned long bAttachedToBall : 1; // 0x03D0 (0x0004)

```

```
[0x0000000000000000] [0x00000008] (CPF_Transient)
struct FInterpCurveFloat           RopeLengthCurve;                                // 0x03D8 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UStaticMesh*                RopeMesh;                                     // 0x03F0 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector                    RopeMeshScale;                                // 0x03F8 (0x000C)
[0x0000000000000001] (CPF_Edit)
float                            RopeMeshInitialSize;                         // 0x0404 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FRotator                  RopeRotationOffset;                           // 0x0408 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FName                      RopeMeshSlackParam;                          // 0x0414 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                      RopeMeshGravityParam;                        // 0x041C (0x0008)
[0x0000000000000001] (CPF_Edit)
class USkeletalMesh*             HookMesh;                                    // 0x0428 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector                   HookMeshScale;                                // 0x0430 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector                   HookMeshOffset;                               // 0x043C (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator                  HookRotationOffset;                           // 0x0448 (0x000C)
[0x0000000000000001] (CPF_Edit)
float                            HitDistanceOffset;                           // 0x0454 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                            AfterAttachDuration;                         // 0x0458 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UAnimSet*                 HookAnimSet;                                // 0x0460 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                     HookClosedAnim;                             // 0x0468 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                     HookOpenAnim;                              // 0x0470 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                            BlockedRequiredMoveDistance;                      // 0x0478 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                            BlockedRequiredMoveTime;                         // 0x047C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                            BlockedStartTime;                           // 0x0480 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FVector                   BlockedStartPos;                            // 0x0484 (0x000C)
[0x0000000000002000] (CPF_Transient)
class UStaticMeshComponent_TA*   RMC;                                       // 0x0490 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UMaterialInstance*        RMI;                                       // 0x0498 (0x0008)
[0x0000000000002000] (CPF_Transient)
class USkeletalMeshComponent_TA* HMC;                                      // 0x04A0 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class ABall_TA*                Ball;                                       // 0x04A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FVector                  RopeOrigin;                                // 0x04B0 (0x000C)
[0x0000000000002000] (CPF_Transient)
float                            RopeToTime;                                // 0x04BC (0x0004)
[0x0000000000002000] (CPF_Transient)
float                            CurrentRopeLength;                         // 0x04C0 (0x0004)
```

```

[0x0000000000000000] (CPF_Transient)
float AttachTime; // 0x04C4 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_GrapplingHook_TA");
}

return uClassPointer;
};

void HandleBallExploded(class ABall_TA* InBall);
void PickupEnd();
void ScaleMeshToLocation(struct FVector NewLocation, struct FVector TargetLocation);
struct FVector GetPredictedBallLocation(class ABall_TA* InBall);
struct FVector GetTargetedLocation();
void HandleCollision(class ARBActor_TA* RB, struct FAccumulatedRigidBodyCollision Collision);
void UpdateVisual(float DeltaTime);
void PickupTick(float DeltaTime);
void ApplyForces(float ActiveTime);
void DoAttach();
void RemovePickupFX();
void InitPickupFX();
void PickupStart();
};

// Class TAGame.SpecialPickup_Spring_TA
// 0x01A8 (0x03C0 - 0x0568)
class ASpecialPickup_Spring_TA : public ASpecialPickup_Targeted_TA
{
public:
float Force; // 0x03C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float VerticalForce; // 0x03C4 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector Torque; // 0x03C8 (0x000C)
[0x0000000000000001] (CPF_Edit)
unsigned long bApplyRelativeForce : 1; // 0x03D4 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bApplyConstantForce : 1; // 0x03D4 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bBreakConstantForceWithHit : 1; // 0x03D4 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long bApplyRelativeConstantForce : 1; // 0x03D4 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
unsigned long bInstant : 1; // 0x03D4 (0x0004)
[0x0000000000000001] [0x00000010] (CPF_Edit)
unsigned long bFollowAfterHit : 1; // 0x03D4 (0x0004)

```

```
[0x0000000000000001] [0x00000020] (CPF_Edit)
unsigned long bSpringed : 1; // 0x03D4 (0x0004)
[0x0000000000000000]
struct FInterpCurveFloat RelativeForceCurve; // 0x03D8 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float RelativeForceNormalDirection; // 0x03F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxSpringLength; // 0x03F4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ConstantForce; // 0x03F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat RelativeConstantForceCurve; // 0x0400
(0x0018) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat SpringLengthCurve; // 0x0418 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FVector FromOffset; // 0x0430 (0x000C)
[0x0000000000000001] (CPF_Edit)
class UStaticMesh* SpringMesh; // 0x0440 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector SpringMeshScale; // 0x0448 (0x000C)
[0x0000000000000001] (CPF_Edit)
float SpringMeshInitialSize; // 0x0454 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FRotator SpringRotationOffset; // 0x0458 (0x000C)
[0x0000000000000001] (CPF_Edit)
class UStaticMesh* HittingMesh; // 0x0468 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector HittingMeshScale; // 0x0470 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector HittingMeshOffset; // 0x047C (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator HittingRotationOffset; // 0x0488 (0x000C)
[0x0000000000000001] (CPF_Edit)
float HitDistanceOffset; // 0x0494 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AfterSpringDuration; // 0x0498 (0x0004)
[0x0000000000000001] (CPF_Edit)
uint8_t BallHitType; // 0x049C (0x0001)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat AfterHitLengthCurve; // 0x04A0 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class AFXActor_TA* HitFXArchetype; // 0x04B8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_TA* ShootFXArchetype; // 0x04C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue* HitSFX; // 0x04C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName FadeOutParam; // 0x04D0 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat FadeOutCurve; // 0x04D8 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float MinSpringLength; // 0x04F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float WeldedForceScalar; // 0x04F4 (0x0004)
```

```

[0x0000000000000001] (CPF_Edit)
float WeldedVerticalForce; // 0x04F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CurrentSpringLength; // 0x04FC (0x0004)
[0x0000000000000000]
float SpringedTime; // 0x0500 (0x0004)
[0x0000000000000000]
class UStaticMeshComponent_TA* SMC; // 0x0508 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UMaterialInstance* SMI; // 0x0510 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UStaticMeshComponent_TA* HMC; // 0x0518 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
TArray<class UMaterialInstance*> HMI; // 0x0520 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
float AfterSpringTime; // 0x0530 (0x0004)
[0x0000000000002000] (CPF_Transient)
float SpringToTime; // 0x0534 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FVector SpringOrigin; // 0x0538 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector SpringedLocation; // 0x0544 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector SpringedNormal; // 0x0550 (0x000C)
[0x0000000000002000] (CPF_Transient)
float SpringedLength; // 0x055C (0x0004)
[0x0000000000002000] (CPF_Transient)
class AFXActor_TA* ShootFX; // 0x0560 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Spring_TA");
}

return uClassPointer;
};

void PickupEnd();
void HandleCarTouchedBall(class ABall_TA* Ball, class ACar_TA* OtherCar, uint8_t HitType);
void ScaleSpringMeshToLocation(struct FVector NewLocation, struct FVector TargetLoction);
void UpdateVisual(float DeltaTime);
void PickupTick(float DeltaTime);
void eventUpdateCarTouchedEvent();
void ApplyForces(float ActiveTime);
struct FVector GetRelativeConstantForce(struct FVector Direction);
struct FVector GetRelativelImpulse(struct FVector Direction);
void SpawnHitFX();
void DoSpring(unsigned long bFirstHit);

```

```

void RemovePickupFX();
void InitPickupFX();
void PickupStart();
};

// Class TAGame.SpecialPickup_BallLasso_TA
// 0x0008 (0x0568 - 0x0570)
class ASpecialPickup_BallLasso_TA : public ASpecialPickup_Spring_TA
{
public:
struct FName SpringMeshSlackParam; // 0x0568 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BallLasso_TA");
}

return uClassPointer;
};

void ScaleSpringMeshToLocation(struct FVector NewLocation, struct FVector TargetLocation);
void DoSpring(unsigned long bFirstHit);
};

// Class TAGame.SpecialPickup_Tornado_TA
// 0x0168 (0x0388 - 0x04F0)
class ASpecialPickup_Tornado_TA : public ASpecialPickup_TA
{
public:
float Height; // 0x0388 (0x0004)
[0x0000000000000001] (CPF_Edit)
float Radius; // 0x038C (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector Offset; // 0x0390 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat VerticalForceCurve; // 0x03A0 (0x0018)
[0x00000000040001] (CPF_Edit | CPF_NeedCtorLink)
float RotationalForce; // 0x03B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat CentripetalForceCurve; // 0x03C0 (0x0018)
[0x00000000040001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat CarDirectionForceCurve; // 0x03D8 (0x0018)
[0x00000000040001] (CPF_Edit | CPF_NeedCtorLink)
float Torque; // 0x03F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector FXScale; // 0x03F4 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector FXOffset; // 0x0400 (0x000C)

```

```

[0x0000000000000001] (CPF_Edit)
class UStaticMesh*           TornadoMesh;          // 0x0410 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector               MeshOffset;          // 0x0418 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector               MeshScale;          // 0x0424 (0x000C)
[0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   TornadoMIC0;        // 0x0430
(0x0008) [0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   TornadoMIC1;        // 0x0438
(0x0008) [0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   TornadoMIC2;        // 0x0440
(0x0008) [0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   TornadoMIC3;        // 0x0448
(0x0008) [0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   TornadoMIC4;        // 0x0450
(0x0008) [0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   TornadoMIC5;        // 0x0458
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat       OpacityCurve;         // 0x0460 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float                         MaxVelocityOffset;      // 0x0478 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                         BallMultiplier;        // 0x047C (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long                 bDebugVis : 1;        // 0x0480 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
float                         VelocityEase;         // 0x0484 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*            BallSFX;            // 0x0488 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*            CarSFX;            // 0x0490 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UStaticMeshComponent*    SMC;                // 0x0498 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UMaterialInstanceConstant*   MIC0;              // 0x04A0 (0x0008)
[0x0000000000000000]
class UMaterialInstanceConstant*   MIC1;              // 0x04A8 (0x0008)
[0x0000000000000000]
class UMaterialInstanceConstant*   MIC2;              // 0x04B0 (0x0008)
[0x0000000000000000]
class UMaterialInstanceConstant*   MIC3;              // 0x04B8 (0x0008)
[0x0000000000000000]
class UMaterialInstanceConstant*   MIC4;              // 0x04C0 (0x0008)
[0x0000000000000000]
class UMaterialInstanceConstant*   MIC5;              // 0x04C8 (0x0008)
[0x0000000000000000]
struct FVector                 Vel;                // 0x04D0 (0x000C)
[0x0000000000002000] (CPF_Transient)
TArray<class ARBActor_TA*>     Affecting;          // 0x04E0 (0x0010)
[0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Tornado_TA");
}

return uClassPointer;
};

void ApplyForces(float ActiveTime);
void RemovePickupFX();
void PlayCarSFX(class ARBActor_TA* InActor);
void PlayBallSFX(class ARBActor_TA* InActor);
void TickPickupFX(float DeltaTime);
void PickupTick(float DeltaTime);
void InitPickupFX();
};

// Class TAGame.CertifiedStat_TA
// 0x0018 (0x0060 - 0x0078)
class UCertifiedStat_TA : public UObject
{
public:
class UClass* StatClass; // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<int32_t> RankThresholds; // 0x0068 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CertifiedStat_TA");
}

return uClassPointer;
};

class FString GetRankLabel(int32_t StatValue);
int32_t GetRank(int32_t StatValue);
class FString GetDescription();
int32_t GetID();
};

// Class TAGame.CertifiedStatDatabase_TA
// 0x0064 (0x0060 - 0x00C4)
class UCertifiedStatDatabase_TA : public UObject
{
public:

```

```

TArray<class UCertifiedStat_TA*>          Stats;           // 0x0060 (0x0010)
[0x0000000000420003] (CPF_Edit | CPF_Const | CPF_EditConst | CPF_NeedCtorLink)
struct FMap_Mirror             StatNameToStatID;        // 0x0070 (0x0050)
[0x00000000000001002] (CPF_Const | CPF_Native)
unsigned long                 bClickToRebuild : 1;      // 0x00C0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CertifiedStatDatabase_TA");
}

return uClassPointer;
};

class UCertifiedStat_TA* GetStatByName(struct FName StatName);
class UCertifiedStat_TA* GetStat(int32_t Id);
struct FName GetStatName(int32_t StatId);
int32_t GetStatId(struct FName StatName);
};

// Class TAGame.ChallengeFolder_TA
// 0x00E0 (0x0060 - 0x0140)
class UChallengeFolder_TA : public UObject
{
public:
class FString           CodeName;           // 0x0060 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString           DisplayName;         // 0x0070 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t                 ChallengeID;        // 0x0080 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                 DefaultActiveIndex;   // 0x0084 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
uint64_t                StartTime;          // 0x0088 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
uint64_t                EndTime;            // 0x0090 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
uint8_t                 ChallengeFolderSort; // 0x0098 (0x0001)
[0x0001000000000000]
TArray<int32_t>         ChallengeIDs;       // 0x00A0 (0x0010)
[0x0001000004000000] (CPF_NeedCtorLink)
TArray<class UChallengeFolder_TA*>    SubFolders;        // 0x00B0
(0x0010) [0x0001000004000000] (CPF_NeedCtorLink)
class FString           SubFolderNames;     // 0x00C0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString           LinkedFolderCodeName; // 0x00D0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
uint64_t                ActiveSubFolderStartTime; // 0x00E0 (0x0008)

```

```

[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t ActiveSubFolderEndTime; // 0x00E8 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
struct FMap_Mirror ChallengeCache; // 0x00F0 (0x0050)
[0x000100000001000] (CPF_Native)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeFolder_TA");
}

return uClassPointer;
};

bool HasValidChallengeID();
class UChallengeFolder_TA* FindFolder(class FString& FindCodeName);
void GetSortedChallenges(class UClass* Filter, TArray<class UChallenge_TA*>& AllChallenges, TArray<class UChallenge_TA*>& Out_SortedChallenges);
void CacheFilterResults(class UClass* Filter, TArray<class UObject*>& Challenges);
bool TryGetFilterResults(class UClass* Filter, TArray<class UChallenge_TA*>& OutResult);
bool TryGetFilterResults_Native(class UClass* Filter, TArray<class UObject*>& OutResult);
void ClearChallengeCache();
void Init();
};

// Class TAGame.ClientInputData_TA
// 0x0024 (0x0060 - 0x0084)
class UClientInputData_TA : public UObject
{
public:
class APlayerController_TA* PC; // 0x0060 (0x0008)
[0x0000000000000000]
TArray<struct FClientFrameData> InputFrames; // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
float LastPacketTimestamp; // 0x0078 (0x0004)
[0x0000000000000000]
int32_t LastProcessedFrame; // 0x007C (0x0004)
[0x0000000000000000]
int32_t LastProcessedReceivedFrame; // 0x0080 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClientInputData_TA");
}

```

```

}

return uClassPointer;
};

};

// Class TAGame.ClientJitterBuffer_TA
// 0x001C (0x0060 - 0x007C)
class UClientJitterBuffer_TA : public UObject
{
public:
class UNetworkJitterSettings_TA* JitterSettings; // 0x0060 (0x0008)
[0x0000800000000000]
float LastPacketTime; // 0x0068 (0x0004)
[0x0000000000000000]
float PacketTimeMu; // 0x006C (0x0004)
[0x0000000000000000]
float PacketTimeVariance; // 0x0070 (0x0004)
[0x0000000000000000]
float BufferTime; // 0x0074 (0x0004)
[0x0000000000000000]
int32_t BufferSize; // 0x0078 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClientJitterBuffer_TA");
}

return uClassPointer;
};

void Reset();
void UpdateBufferSize();
void ReceivedPacket();
};

// Class TAGame.CollisionTestComponent_TA
// 0x004F (0x009D - 0x00EC)
class UCollisionTestComponent_TA : public UActorComponent
{
public:
class AActor* LineStart; // 0x00A0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AActor* LineEnd; // 0x00A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AActor* ArchetypeToSpawn; // 0x00B0 (0x0008)
[0x0000000000000001] (CPF_Edit)

```

```
struct FVector           CachedCenter;          // 0x00B8 (0x000C)
[0x0000000000000002] (CPF_Const)
struct FVector           CachedStart;          // 0x00C4 (0x000C)
[0x0000000000000002] (CPF_Const)
struct FVector           CachedEnd;           // 0x00D0 (0x000C)
[0x0000000000000002] (CPF_Const)
float                   Speed;                // 0x00DC (0x0004)
[0x0000000000000001] (CPF_Edit)
float                   Step;                 // 0x00E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                   Time;                // 0x00E4 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long           bEnabled : 1;        // 0x00E8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CollisionTestComponent_TA");
}

return uClassPointer;
};

// Class TAGame.ContentConfig_TA
// 0x0000 (0x0088 - 0x0088)
class UContentConfig_TA : public UContentConfig_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ContentConfig_TA");
}

return uClassPointer;
};

void Apply();
void UpdateAvailableContent();
};

// Class TAGame.CrossEntitlementOverrideConfig_TA
```

```

// 0x0010 (0x0078 - 0x0088)
class UCrossEntitlementOverrideConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FXEStatusOverride> XEStatusOverrides; // 0x0078
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrossEntitlementOverrideConfig_TA");
}

return uClassPointer;
};

uint8_t GetBaseXEStatus(class UProduct_TA* Product);
uint8_t GetBaseXEStatusFromID(int32_t ProductID);
void ProcessXEOverride(struct FXEStatusOverride StatusOverride);
void Apply();
};

// Class TAGame.CrowdActorSettings_TA
// 0x0118 (0x0060 - 0x0178)
class UCrowdActorSettings_TA : public UObject
{
public:
float NoiseRiseRate; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
float NoiseFallRate; // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FRandomRange IdleNoise; // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FRandomRange IdleNoiseDuration; // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UCrowdNoiseModifier_TA* CountdownStartNoise; // 0x0078
(0x0008) [0x0000000004000001] (CPF_Edit | CPF_EditInline)
class UCrowdNoiseModifier_TA* CountdownEndNoise; // 0x0080
(0x0008) [0x0000000004000001] (CPF_Edit | CPF_EditInline)
class UCrowdNoiseModifier_TA* OvertimeStartNoise; // 0x0088
(0x0008) [0x0000000004000001] (CPF_Edit | CPF_EditInline)
class UCrowdNoiseModifier_TA* TimeAlmostOutNoise; // 0x0090
(0x0008) [0x0000000004000001] (CPF_Edit | CPF_EditInline)
TArray<struct FStatCrowdNoise> StatNoises; // 0x0098 (0x0010)
[0x000000000400041] (CPF_Edit | CPF_EditConstArray | CPF_NeedCtorLink)
class UAkSoundCue* RandomChantsSound; // 0x00A8
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FRandomRange RandomChantsDelay; // 0x00B0
(0x0008) [0x0000000000000001] (CPF_Edit)
unsigned long UpdateStatNoises : 1; // 0x00B8 (0x0004)

```

```

[0x0000000800000001] [0x00000001] (CPF_Edit)
struct FInterpCurveFloat           BallGoalDistanceNoiseCurve;          // 0x00C0
(0x0018) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FVector                   GoalDotDirection;                  // 0x00D8 (0x00C)
[0x0000000000000001] (CPF_Edit)
float                           MinGoalDotAngle;                // 0x00E4 (0x004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FBallHitCrowdNoise> BallHitNoises;                 // 0x00E8
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float                           BallHitNoiseDelay;               // 0x00F8 (0x004)
[0x0000000000000001] (CPF_Edit)
TArray<class UAkSoundCue*>      CountDownSoundList;            // 0x0100
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UAkSoundCue*>      RoundCountDownSoundList;        // 0x0110
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UAkSoundCue*              OneMinRemaining;               // 0x0120 (0x008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*              ThirtySecondsRemaining;        // 0x0128
(0x0008) [0x0000000000000001] (CPF_Edit)
class UAkSoundCue*              CrowdStingerWin;              // 0x0130 (0x008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*              CrowdStingerLose;             // 0x0138 (0x008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*              CrowdStingerForfeit;          // 0x0140 (0x008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*              CrowdStartOverTimeSound;       // 0x0148
(0x0008) [0x0000000000000001] (CPF_Edit)
class UAkSoundCue*              MatchEndedEncore;             // 0x0150 (0x008)
[0x0000000000000001] (CPF_Edit)
float                           MatchEndedEncoreDelay;          // 0x0158 (0x004)
[0x0000000000000001] (CPF_Edit)
struct FCrowdDefenseSettings    DefenseChant;                 // 0x0160 (0x0018)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdActorSettings_TA");
}

return uClassPointer;
};

};

// Class TAGame.CrowdNoiseModifier_TA
// 0x0018 (0x0060 - 0x0078)
class UCrowdNoiseModifier_TA : public UObject
{
public:

```

```
float           ValueMin;           // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
float           ValueMax;           // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)
float           DurationMin;        // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)
float           DurationMax;        // 0x006C (0x0004)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*      OneShotSound; // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdNoiseModifier_TA");
}

return uClassPointer;
};

};

// Class TAGame.DataAsset_ESportsTeam_TA
// 0x0010 (0x0060 - 0x0070)
class UDataAsset_ESportsTeam_TA : public UObject
{
public:
int32_t          TeamID;           // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UProductAssetReference_TA*    ESportsTeamReference; // 0x0068
(0x0008) [0x000000004000001] (CPF_Edit | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DataAsset_ESportsTeam_TA");
}

return uClassPointer;
};

};

// Class TAGame.DataAssetDatabase_TA
// 0x006C (0x0060 - 0x00CC)
class UDataAssetDatabase_TA : public UObject
```

```

{
public:
class UClass* DataAssetClass; // 0x0060 (0x0008)
[0x0000000000020001] (CPF_Edit | CPF_EditConst)
TArray<class UObject*> DataAssets; // 0x0068 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FMap_Mirror NameTold; // 0x0078 (0x0050)
[0x0000000000001002] (CPF_Const | CPF_Native)
unsigned long bClickToRebuild : 1; // 0x00C8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DataAssetDatabase_TA");
}

return uClassPointer;
};

class UObject* GetAssetFromName(class UClass* ObjClass, struct FName DataAssetName);
class UObject* GetAssetFromID(class UClass* ObjClass, int32_t DataAssetID);
struct FName GetName(int32_t DataAssetID);
int32_t GetID(struct FName DataAssetName);
};

// Class TAGame.DataAssetDatabase_ESportsTeam_TA
// 0x0004 (0x00CC - 0x00D0)
class UDataAssetDatabase_ESportsTeam_TA : public UDataAssetDatabase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DataAssetDatabase_ESportsTeam_TA");
}

return uClassPointer;
};

};

// Class TAGame.DecalComponent_Constrained_TA
// 0x0008 (0x03E0 - 0x03E8)
class UDecalComponent_Constrained_TA : public UDecalComponent

```

```
{  
public:  
    float          WorldZ;           // 0x03E0 (0x0004)  
    [0x0000000000000001] (CPF_Edit)  
    unsigned long   bAttachedRotation : 1; // 0x03E4 (0x0004)  
    [0x0000000000000001] [0x00000001] (CPF_Edit)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.DecalComponent_Constrained_TA");  
        }  
  
        return uClassPointer;  
    };  
};  
  
};  
  
// Class TAGame.DistributionFloatSpeed_TA  
// 0x0014 (0x00A1 - 0x00B5)  
class UDistributionFloatSpeed_TA : public UDistributionFloatParameterBase  
{  
public:  
    struct FVector      VelocityScale; // 0x00A8 (0x000C)  
    [0x0000000000000001] (CPF_Edit)  
    uint8_t             VelocitySpace; // 0x00B4 (0x0001)  
    [0x0000000000000001] (CPF_Edit)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.DistributionFloatSpeed_TA");  
        }  
  
        return uClassPointer;  
    };  
};  
  
};  
  
// Class TAGame.DistributionFloatSteer_TA  
// 0x0007 (0x00A1 - 0x00A8)  
class UDistributionFloatSteer_TA : public UDistributionFloatParameterBase  
{  
public:  
  
public:
```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DistributionFloatSteer_TA");
}

return uClassPointer;
};

};

// Class TAGame.DLCPack_TA
// 0x0080 (0x0060 - 0x00E0)
class UDLCPack_TA : public UObject
{
public:
TArray<uint8_t> UnlockedPlatforms; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<uint8_t> DiscUnlockedPlatforms; // 0x0070 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UProductSlot_TA* LabelSlot; // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t SteamID; // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)
class FString PS4ID; // 0x0090 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString XboxOneID; // 0x00A0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString SwitchID; // 0x00B0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString EpicId; // 0x00C0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UProduct_TA*> Products; // 0x00D0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DLCPack_TA");
}

return uClassPointer;
};

bool IsEnabledOnPlatform(uint8_t InOnlinePlatform);
class FString GetDLCID(uint8_t InOnlinePlatform);
int32_t GetDLCLabelSlotProductID(int32_t SubIdx);

```

```

};

// Class TAGame.EngineAudioBaseComponent_TA
// 0x005C (0x00A4 - 0x0100)
class UEngineAudioBaseComponent_TA : public UActorComponent_X
{
public:
    class UAkPlaySoundComponent*           EngineAudio;          // 0x00A8
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UAkPlaySoundComponent*           ExhaustAudio;        // 0x00B0
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UEngineAudioBlowoffComponent_TA* BlowoffComponent;    // 0x00B8 (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UWheelSpeedComponent_TA*        WheelSpeed;          // 0x00C0
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UThrottleStateComponent_TA*     ThrottleComponent;   // 0x00C8
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UAkSoundCue*                  GearChangeSound;      // 0x00D0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UMaxActorsGroup_TA*           MaxActorsGroup;       // 0x00D8
    (0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    class AVehicle_TA*                 Car;                  // 0x00E0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    struct FScriptDelegate             __GetNormalizedRpmDelegate__Delegate; // 0x00E8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.EngineAudioBaseComponent_TA");
        }

        return uClassPointer;
    };

    void PrintDebugInfo(class UDebugDrawer* Drawer);
    void eventGearChanged();
    void HandleSplitScreenTypeChanged(class UGameViewportClient_TA* ViewportClient);
    void HandleDisabled();
    void HandleEnabled();
    void InitFromAsset(class UProductAsset_EngineAudio_TA* EngineAsset);
    void eventDetached();
    void eventAttached();
    float GetNormalizedRpmDelegate();
}

```

```

};

// Class TAGame.EngineAudioComponent_TA
// 0x0020 (0x0100 - 0x0120)
class UEngineAudioComponent_TA : public UEngineAudioBaseComponent_TA
{
public:
    class UEngineAudioProfile_TA*           Profile;           // 0x0100 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    struct FCarStateData                  OldCarState;      // 0x0108 (0x0010)
    [0x0000000000002000] (CPF_Transient)
    struct FPointer                      NativeState;      // 0x0118 (0x0008)
    [0x0000000000003002] (CPF_Const | CPF_Native | CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.EngineAudioComponent_TA");
        }

        return uClassPointer;
    };

    void PrintDebugInfo(class UDebugDrawer* Drawer);
    float GetNormalizedRpm();
    struct FName GetNativeStateName();
    void DestroyNativeState();
    void HandleDisabled();
    void InitFromAsset(class UProductAsset_EngineAudio_TA* EngineAsset);
};

// Class TAGame.EngineAudioREVComponent_TA
// 0x00A0 (0x0100 - 0x01A0)
class UEngineAudioREVComponent_TA : public UEngineAudioBaseComponent_TA
{
public:
    class UEngineAudioProfileREV_TA*       Profile;           // 0x0100 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkRevPhysicsSimulation*        PhysicsSim;       // 0x0108 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UEngineAudioRevSimLimiter_TA*    RevLimiter;       // 0x0110
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UEngineAudioRevSimReverse_TA*    Reverse;          // 0x0118
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UEngineAudioRevSimVelocitySync_TA* VelocitySync;   // 0x0120
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UEngineAudioRevSimBoost_TA*      Boost;           // 0x0128 (0x0008)
};

```

```

[0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UEngineAudioRevSimBoostLevel_TA*      BoostLevel;           // 0x0130
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF>EditInline)
class UEngineAudioRevSimResistance_TA*      Resistance;          // 0x0138
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF>EditInline)
class UEngineAudioRevSimGearManager_TA*     GearManager;         // 0x0140
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
struct FAkRevSimFrame                      SimFrame;           // 0x0148 (0x0010)
[0x0000000000002000] (CPF_Transient)
struct FAkRevSimUpdateParams                CachedUpdateParams; // 0x0158
(0x0048) [0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioREVComponent_TA");
}

return uClassPointer;
};

float __EngineAudioREVComponent_TA__InitFromAsset_0x2();
void __EngineAudioREVComponent_TA__InitFromAsset_0x1(class UAkRevPhysicsSimulation* S);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void InitFromAsset(class UProductAsset_EngineAudio_TA* EngineAsset);
};

// Class TAGame.EngineAudioBlowoffComponent_TA
// 0x0020 (0x00A4 - 0x00C4)
class UEngineAudioBlowoffComponent_TA : public UActorComponent_X
{
public:
class UAkSoundCue*                      BlowoffSound;        // 0x00A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                                 BlowoffThrottleTime; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AVehicle_TA*                     Car;                // 0x00B8 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                                 FullThrottleTime; // 0x00C0 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.EngineAudioBlowoffComponent_TA");
}

return uClassPointer;
};

void eventAttached();
};

// Class TAGame.EngineAudioPreviewBase_TA
// 0x0024 (0x0060 - 0x0084)
class UEngineAudioPreviewBase_TA : public UObject
{
public:
    float StartDelay; // 0x0060 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float AccelerateDelay; // 0x0064 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float AccelerateLength; // 0x0068 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float IdleLength; // 0x006C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UProductAsset_EngineAudio_TA* EngineAsset; // 0x0070
    (0x0008) [0x0000000000000000]
    class UAkParamGroup* Ak; // 0x0078 (0x0008)
    [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
    float Throttle; // 0x0080 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.EngineAudioPreviewBase_TA");
        }
    }

    return uClassPointer;
};

void GotoPreviewState();
void eventGearChanged(int32_t GearNum);
void Update(float DeltaTime);
void StartAccelerating();
void StopAccelerating();
void StopPreview();
void StartPreview();
void Init(class UProductAsset_EngineAudio_TA* InAsset, class UAkParamGroup* InAkParams);
};

// Class TAGame.EngineAudioPreview_TA
// 0x0044 (0x0084 - 0x00C8)

```

```

class UEngineAudioPreview_TA : public UEngineAudioPreviewBase_TA
{
public:
class UEngineAudioProfile_TA*           Profile;           // 0x0088 (0x0008)
[0x0000000000000000]
struct FPointer                         NativeState;      // 0x0090 (0x0008)
[0x000000000001002] (CPF_Const | CPF_Native)
struct FInterpCurveFloat                 GearToWheelSpeedAccel; // 0x0098
(0x0018) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float                                 PreviewSpeed;     // 0x00B0 (0x0004)
[0x0000000000000000]
float                                 MaxPreviewSpeed; // 0x00B4 (0x0004)
[0x0000000000000000]
float                                 WheelSpeedToPreviewSpeed; // 0x00B8 (0x0004)
[0x0000000000000000]
float                                 BrakeFactor;      // 0x00BC (0x0004)
[0x0000000000000000]
struct FName                           IsLocalRTPC;    // 0x00C0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioPreview_TA");
}

return uClassPointer;
};

void eventGearChanged(int32_t GearNum);
void DestroyNativeState();
void UpdateNativeState(float DeltaTime);
void Init(class UProductAsset_EngineAudio_TA* InAsset, class UAkParamGroup* InAkParams);
};

// Class TAGame.EngineAudioProfileBase_TA
// 0x0010 (0x0060 - 0x0070)
class UEngineAudioProfileBase_TA : public UObject
{
public:
float                                 BlowoffThrottleTime; // 0x0060 (0x0004)
[0x0000000000000001] (CPF>Edit)
float                                 WheelForwardSpeedInterpRate; // 0x0064 (0x0004)
[0x0000000000000001] (CPF>Edit)
float                                 WheelSideSpeedInterpRate; // 0x0068 (0x0004)
[0x0000000000000001] (CPF>Edit)
float                                 MaxWheelSpeed;      // 0x006C (0x0004)
[0x0000000000000001] (CPF>Edit)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioProfileBase_TA");
}

return uClassPointer;
};

};

// Class TAGame.EngineAudioProfile_TA
// 0x0050 (0x0070 - 0x00C0)
class UEngineAudioProfile_TA : public UEngineAudioProfileBase_TA
{
public:
TArray<struct FAudioGear> Gears; // 0x0070 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float GearSwitchTime; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMAccelStart; // 0x0084 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMAccelEnd; // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMAccelClutched; // 0x008C (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMDecelClutched; // 0x0090 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMMaxClutched; // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMAccelFactor; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMDecelFactor; // 0x009C (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMAccelBoostStart; // 0x00A0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMAccelBoostEnd; // 0x00A4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMShiftUpBoost; // 0x00A8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RPMMaxLoad; // 0x00AC (0x0004)
[0x0000000000000001] (CPF_Edit)
float EngineLoadSmoothFactor; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AirMaxThrottleTime; // 0x00B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RevLimitRPM; // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RevLimitRPMDecel; // 0x00BC (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioProfile_TA");
}

return uClassPointer;
};

};

// Class TAGame.EngineAudioProfileREV_TA
// 0x0030 (0x0070 - 0x00A0)
class UEngineAudioProfileREV_TA : public UEngineAudioProfileBase_TA
{
public:
struct FAkRevSimPhysicsControls           Controls;           // 0x0070 (0x0020)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UAkRevPhysicsSimulation*           PhysicsSim;        // 0x0090 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UEngineAudioRevSimLimiter_TA*       RevLimiter;        // 0x0098
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioProfileREV_TA");
}

return uClassPointer;
};

};

// Class TAGame.EngineAudioRevSimBoost_TA
// 0x0010 (0x0070 - 0x0080)
class UEngineAudioRevSimBoost_TA : public UComponent
{
public:
float             InterpTime;           // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float             InterpScale;          // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
float             InterpExp;            // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```
float ActiveTime; // 0x007C (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimBoost_TA");
}

return uClassPointer;
};

// Class TAGame.EngineAudioRevSimBoostLevel_TA
// 0x0020 (0x0070 - 0x0090)
class UEngineAudioRevSimBoostLevel_TA : public UComponent
{
public:
float ActivatedInterpTime; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float InactiveInterpTime; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
float InterpExp; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ActivatedSpeedSq; // 0x007C (0x0004)
[0x0000000000002000] (CPF_Transient)
float InterpTime; // 0x0080 (0x0004)
[0x0000000000002000] (CPF_Transient)
float InterpValue; // 0x0084 (0x0004)
[0x0000000000002000] (CPF_Transient)
float InterpTimeElapsed; // 0x0088 (0x0004)
[0x0000000000002000] (CPF_Transient)
float TargetValue; // 0x008C (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimBoostLevel_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
```

```

};

// Class TAGame.EngineAudioRevSimGearManager_TA
// 0x0014 (0x0070 - 0x0084)
class UEngineAudioRevSimGearManager_TA : public UComponent
{
public:
int32_t RealGear; // 0x0070 (0x0004)
[0x0000004000000000]
int32_t FakeGear; // 0x0074 (0x0004)
[0x0000004000000000]
unsigned long bPendingSilentDownShift : 1; // 0x0078 (0x0004)
[0x0000000000000000] [0x00000001]
float TimeRemaining; // 0x007C (0x0004)
[0x0000004000000000]
float SilentDownShiftDelay; // 0x0080 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimGearManager_TA");
}

return uClassPointer;
};

void TickGear(float DeltaTime, struct FAkRevSimFrame& SimFrame, struct
FAkRevSimUpdateParams& SimUpdate);
};

// Class TAGame.EngineAudioRevSimLimiter_TA
// 0x0018 (0x0070 - 0x0088)
class UEngineAudioRevSimLimiter_TA : public UComponent
{
public:
float LimitTime; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideSpeedThresh; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DecelScale; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AirMaxThrottleTime; // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
float TimeRemaining; // 0x0080 (0x0004)
[0x0000004000002000] (CPF_Transient)
float TimeActiveInAir; // 0x0084 (0x0004)
[0x0000004000002000] (CPF_Transient)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimLimiter_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
};

// Class TAGame.EngineAudioRevSimResistance_TA
// 0x002C (0x0070 - 0x009C)
class UEngineAudioRevSimResistance_TA : public UComponent
{
public:
float Clutched; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ZScale; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat SideSpeedCurve; // 0x0078 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float SideFriction; // 0x0090 (0x0004)
[0x0000000000002000] (CPF_Transient)
float ZFriction; // 0x0094 (0x0004)
[0x0000000000002000] (CPF_Transient)
float SideAngle; // 0x0098 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimResistance_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
};

// Class TAGame.EngineAudioRevSimReverse_TA
// 0x0004 (0x0070 - 0x0074)
class UEngineAudioRevSimReverse_TA : public UComponent
{
public:

```

```
float ReverseGearScale; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimReverse_TA");
}

return uClassPointer;
};

};

// Class TAGame.EngineAudioRevSimVelocitySync_TA
// 0x001C (0x0070 - 0x008C)
class UEngineAudioRevSimVelocitySync_TA : public UComponent
{
public:
float NoThrottleTime; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float NoThrottleTimeElapsed; // 0x0074 (0x0004)
[0x0000000000002000] (CPF_Transient)
float RealToSimScale; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float InterpSpeed; // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
float InterpTime; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
float InterpTimeLeft; // 0x0084 (0x0004)
[0x0000000000002000] (CPF_Transient)
float FirstGearThrottleThreshold; // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioRevSimVelocitySync_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
};
```

```

// Class TAGame.EngineShare_TA
// 0x00D8 (0x0170 - 0x0248)
class UEngineShare_TA : public UEngineShare_X
{
public:
    unsigned long          bEnableClientPrediction : 1;           // 0x0170 (0x0004)
    [0x0000000000000003] [0x00000001] (CPF_Edit | CPF_Const)
    unsigned long          bClientPhysicsUpdate : 1;           // 0x0170 (0x0004)
    [0x0000000000002002] [0x00000002] (CPF_Const | CPF_Transient)
    unsigned long          bDisableClientCorrections : 1;        // 0x0170 (0x0004)
    [0x0000000000002000] [0x00000004] (CPF_Transient)
    unsigned long          bRunningPhysicsFrame : 1;           // 0x0170 (0x0004)
    [0x0000000000000002] [0x00000008] (CPF_Const)
    unsigned long          bDebugClientCorrections : 1;         // 0x0170 (0x0004)
    [0x0000000000000000] [0x00000010]
    unsigned long          bForceClientCorrection : 1;         // 0x0170 (0x0004)
    [0x0000000000000000] [0x00000020]
    class UReplayManager_TA*      ReplayManager;                // 0x0178 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    class UAdManager_TA*       AdManager;                    // 0x0180 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    class UOnlineProductStore_TA*   OnlineProductStore;      // 0x0188
    (0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    float                  PhysicsFramerate;             // 0x0190 (0x0004)
    [0x0000000000000001] (CPF>Edit)
    int32_t                MaxPhysicsSubsteps;            // 0x0194 (0x0004)
    [0x0000000000000003] (CPF>Edit | CPF_Const)
    int32_t                MaxUploadedClientFrames;        // 0x0198 (0x0004)
    [0x0000000000000001] (CPF>Edit)
    int32_t                MaxClientReplayFrames;          // 0x019C (0x0004)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    int32_t                PhysicsFrame;                 // 0x01A0 (0x0004)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    float                  RenderAlpha;                 // 0x01A4 (0x0004)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    int32_t                ReplicatedPhysicsFrame;        // 0x01A8 (0x0004)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    int32_t                DirtyPhysicsFrame;            // 0x01AC (0x0004)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    int32_t                ForceCorrectionFrames;         // 0x01B0 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    TArray<class UITickNotify_TA*>   TickNotifies;        // 0x01B8 (0x0010)
    [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
    int32_t                TickNotifyIndex;              // 0x01C8 (0x0004)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    class FString          ShellArchetypePath;           // 0x01D0 (0x0010)
    [0x0000000000404000] (CPF_Config | CPF_NeedCtorLink)
    class USystemSettingsManager_TA*   SystemSettingsManager; // 0x01E0
    (0x0008) [0x0000000000002000] (CPF_Transient)
    float                  DebugClientCorrectionStartTime; // 0x01E8 (0x0004)
    [0x0000000000000000]
    int32_t                DebugClientCorrectionCount;    // 0x01EC (0x0004)
    [0x0000000000000000]

```

```
class UStatGraphSystem_TA*           StatGraphs;          // 0x01F0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPhysicsConfig_TA*           PhysicsConfig;       // 0x01F8 (0x0008)
[0x0000800000000000]
TArray<class ANetworkInputBuffer_TA*> InputBuffers;      // 0x0200
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)
float                             LastPhysicsDeltaTimeScale; // 0x0210 (0x0004)
[0x0000000000000000]
struct FScriptDelegate             __EventPostPhysicsStep__Delegate; // 0x0218
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __EventPreAsyncTick__Delegate;   // 0x0230
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineShare_TA");
}

return uClassPointer;
};

void AddDynamicallyLoadedPackages(TArray<struct FDynamicallyLoadedPackage>&
OutPackages);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void eventOnReliablePacketsBlocked();
void DebugClientCorrections();
float GetBulletFixedDeltaTime();
void RunPhysicsStep(int32_t BulletSceneIndex, float DeltaTime);
void DrawBuildInfo(class UCanvas* C);
void UpdateReplicatedPhysicsFrame(int32_t ServerFrame);
static void DebugDedicatedServer(float ForHowLong);
float GetPhysicsTime();
void SetTickNotify(class UTickNotify_TA* Obj, unsigned long bNotify);
static class UObject* LoadStandaloneObject(class UClass* ExpectedClass, struct FName
ObjectName, int32_t LoadFlags);
void InitOnlineGame(class UOnlineSubsystem* NewOnlineSubsystem);
void eventRecordAppStart();
void eventInit();
static class FString GetVersionString();
void EventPreAsyncTick(float DeltaTime);
void EventPostPhysicsStep(float DeltaTime);
};

// Class TAGame.Errors_TA
// 0x00D8 (0x0080 - 0x0158)
class UErrors_TA : public UErrorList
{
public:
class UErrorType*                 ReplayImport;        // 0x0080 (0x0008)
```

```
[0x0000000000000002] (CPF_Const)
class UErrorType*           ReplayExport;          // 0x0088 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           ReplayNoReplay;        // 0x0090 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           ReplayExportNoSpace;   // 0x0098 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           UpdateRequired;       // 0x00A0 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TradeInMaxed;         // 0x00A8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           OrderNotFound;        // 0x00B0 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           InventoryOutOfSync;   // 0x00B8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TimedOut;             // 0x00C0 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingModeNotFound; // 0x00C8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingModeDownloadFailed; // 0x00D0
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingModelInvalidFavoritedCodes; // 0x00D8
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingModeListPlayerTrainingFailed; // 0x00E0
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingModeAddFailed;   // 0x00E8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingModeUpdateFailed; // 0x00F0 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TrainingNameInvalid;    // 0x00F8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           CodeIsNotValid;        // 0x0100 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           CodeHasBeenRedeemed;   // 0x0108 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           CodeMaxIncorrectAttempts; // 0x0110 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           PlayerAlreadyLinked;   // 0x0118 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           InvalidPlayerPlatformAuth; // 0x0120 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           OnlineAchievementsSaveDataNotFound; // 0x0128
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           ShopItemPlayerPurchaseMax; // 0x0130
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           MicrotransactionProductPurchaseLimitReached; // 0x0138
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           TwoFactorAuthenticationRequired; // 0x0140
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           MatchHasConcluded;      // 0x0148 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           InvalidRocketPassPurchase; // 0x0150 (0x0008)
[0x0000000000000002] (CPF_Const)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Errors_TA");
}

return uClassPointer;
};

};

// Class TAGame.ExplosionHitHandler_TA
// 0x005C (0x00F4 - 0x0150)
class UExplosionHitHandler_TA : public UExplosionHitHandler_X
{
public:
unsigned long          bApplyToAllCars : 1;           // 0x00F8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bIsActive : 1;                // 0x00F8 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
struct FMap_Mirror      VehicleToContactInfo;        // 0x0100 (0x0050)
[0x0000000000003000] (CPF_Native | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ExplosionHitHandler_TA");
}

return uClassPointer;
};

struct FExplosionPaintInfo eventGetExplosionPaintInfo();
void DrawDebugLine(struct FColor LineColor, struct FVector& Start, struct FVector& End);
void DrawDebugSphere(float Radius, struct FColor SphereColor, struct FVector& Location);
void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
void eventProcessHit(class AActor* Victim, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

// Class TAGame.DefaultHitHandler_TA
// 0x0004 (0x0150 - 0x0154)
class UDefaultHitHandler_TA : public UExplosionHitHandler_TA
{
public:

```

```

float ImpulseVelocityMultiplier; // 0x0150 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DefaultHitHandler_TA");
}

return uClassPointer;
};

struct FVector GetImpulseDirection(struct FVector& HitLocation, struct FVector& ImpulseLocation, struct FRotator& ImpulseRotation);
void ApplyImpulseTowardDirection(class ACar_TA* CarHit, float DamageScale, struct FVector& HitLocation, struct FVector& ImpulseLocation, struct FVector& ImpulseDirection);
void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

// Class TAGame.OrbitHitHandler_TA
// 0x0074 (0x0154 - 0x01C8)
class UOrbitHitHandler_TA : public UDefaultHitHandler_TA
{
public:
struct FVector RotationAxis; // 0x0158 (0x000C)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float CarDistanceBeforeOrbit; // 0x0164 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float StartAffectedRadius; // 0x0168 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float EndAffectedRadius; // 0x016C (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float StartOrbitDistance; // 0x0170 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float EndOrbitDistance; // 0x0174 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float StartOrbitSpeed; // 0x0178 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float EndOrbitSpeed; // 0x017C (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float FarAwayOrbitSpeedMultiplier; // 0x0180 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float MaxDeactivationVelocity; // 0x0184 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
uint8_t OrbitCenterLocation; // 0x0188 (0x0001)
[0x0000000000000003] (CPF_Edit | CPF_Const)
uint8_t OrbitAxisBehavior; // 0x0189 (0x0001)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FColor AffectedAreaColor; // 0x018C (0x0004)

```

```

[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FColor OrbitCenterColor; // 0x0190 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FColor OrbitLocationColor; // 0x0194 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
TArray<struct FOrbitCarData> OrbitingCars; // 0x0198 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FVector OrbitCenter; // 0x01A8 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector NormalizedRotationAxis; // 0x01B4 (0x000C)
[0x0000000000002000] (CPF_Transient)
float AffectedRadiusSquared; // 0x01C0 (0x0004)
[0x0000000000002000] (CPF_Transient)
float OrbitDistance; // 0x01C4 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OrbitHitHandler_TA");
}

return uClassPointer;
};

void UpdateOrbitData(float OrbitSpeed, float DeltaTime, struct FOrbitCarData& CarData);
int32_t RegisterNewCarAndGetIndex(class AVehicle_TA* CarHit, struct FVector& HitLocation);
void Init(class UActorComponent_X* ExplosionGoal);
void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

// Class TAGame.GreenScreenEffectHandler_TA
// 0x0018 (0x0150 - 0x0168)
class UGreenScreenEffectHandler_TA : public UExplosionHitHandler_TA
{
public:
struct FColor ColorToApply; // 0x0150 (0x0004)
[0x0000000000000001] (CPF_Edit)
float FadeInDuration; // 0x0154 (0x0004)
[0x0000000000000001] (CPF_Edit)
float FadeOutDuration; // 0x0158 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bUseSystemSettings : 1; // 0x015C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class AGreenScreen_TA* GreenScreenActor; // 0x0160 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GreenScreenEffectHandler_TA");
}

return uClassPointer;
};

void eventDetached();

// Class TAGame.JiggleHitHandler_TA
// 0x0038 (0x0150 - 0x0188)
class UJiggleHitHandler_TA : public UExplosionHitHandler_TA
{
public:
float JiggleDistance; // 0x0150 (0x0004)
[0x0000000000000001] (CPF_Edit)
float JiggleSpeed; // 0x0154 (0x0004)
[0x0000000000000001] (CPF_Edit)
float JiggleSlowDownSpeed; // 0x0158 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UParticleSystem* BounceOffParticleSystem; // 0x0160
(0x0008) [0x0000000000000001] (CPF_Edit)
TArray<struct FJiggleCarData> JigglingCars; // 0x0168 (0x0010)
[0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class AActor*> SpawnedParticleActors; // 0x0178 (0x0010)
[0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.JiggleHitHandler_TA");
}

return uClassPointer;
};

void RecordOrBounceNewCar(class AVehicle_TA* CarHit, struct FVector& HitLocation);
void Destroy();
void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

// Class TAGame.OverrideMaterialsHitHandler_TA
// 0x0030 (0x0150 - 0x0180)
class UOverrideMaterialsHitHandler_TA : public UExplosionHitHandler_TA

```

```

{
public:
uint8_t OverrideMaterialMode; // 0x0150 (0x0001)
[0x0000000000000001] (CPF_Edit) class UMaterialInterface* MaterialToUse; // 0x0158 (0x0008)
[0x0000000000000001] (CPF_Edit) class UMaterialInterface* PaintedMaterialToUse; // 0x0160 (0x0008)
[0x0000000000000001] (CPF_Edit) class UMaterialInterface* BallMaterialToUse; // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit) TArray<struct FSkeletalMeshMaterialCache> CachedSkeletalMeshes; // 0x0170 (0x0010) [0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OverrideMaterialsHitHandler_TA");
}

return uClassPointer;
};

void eventApplyExplosionPaintToMesh(class UMeshComponent* InMeshComponent, struct FExplosionPaintInfo& PaintInfo);
class ABall_TA* eventFindFirstGameBall();
void eventProcessHit(class AActor* Victim, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
void CacheAndOverrideMaterialsForCarMesh(class UCarMeshComponent_TA* CarMesh);
};

// Class TAGame.FeatureFilterTest_TA
// 0x0000 (0x0060 - 0x0060)
class UFeatureFilterTest_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FeatureFilterTest_TA");
}

return uClassPointer;
};

```

```

int32_t GetImplementationMacro();
void Test();
};

// Class TAGame.FlockActor_TA
// 0x006C (0x0268 - 0x02D4)
class AFlockActor_TA : public AActor
{
public:
    class UStaticMesh*           AgentMesh;           // 0x0268 (0x0008)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    class UInstancedStaticMeshComponent*   FlockInstancedMesh;       // 0x0270
    (0x0008) [0x00000000408000A] (CPF_Const | CPF_ExportObject | CPF_Component | 
    CPF_EditInline)
    int32_t                 FlockSize;             // 0x0278 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                   SeparationWeight;        // 0x027C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FVector          FlockRadius;          // 0x0280 (0x000C)
    [0x0000000200000001] (CPF_Edit)
    float                   AgentSpeed;           // 0x028C (0x0004)
    [0x0000000200000001] (CPF_Edit)
    float                   SeparationIdealDistance; // 0x0290 (0x0004)
    [0x0000000200000001] (CPF_Edit)
    class ASplineActor*     Spline;               // 0x0298 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float                   SpawnSpreadDistanceAlongSpline; // 0x02A0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    unsigned long           bRestartTimerEnabled : 1;      // 0x02A4 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long           UpdateFlockMovement : 1;      // 0x02A4 (0x0004)
    [0x0000000200000001] [0x00000002] (CPF_Edit)
    unsigned long           ResetFlockOnToggleUpdateMovement : 1; // 0x02A4
    (0x0004) [0x0000000200000001] [0x00000004] (CPF_Edit)
    float                   WaitTimeInSeconds;         // 0x02A8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                   MinAgentScale;          // 0x02AC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                   MaxAgentScale;          // 0x02B0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                   CurrentRestartTime;       // 0x02B4 (0x0004)
    [0x0000000000000000]
    TArray<struct FFlockAgent> Agents;           // 0x02B8 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FVector          AverageAgentPosition;    // 0x02C8 (0x000C)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {

```

```

uClassPointer = UObject::FindClass("Class TAGame.FlockActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.ForceVolume_TA
// 0x0024 (0x02A4 - 0x02C8)
class AForceVolume_TA : public AVolume
{
public:
    struct FPointer           VfTable_IITickNotify_TA;          // 0x02A8 (0x0008)
    [0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
    uint8_t                   ForceDirection;                  // 0x02B0 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    uint8_t                   ConstantForceMode;            // 0x02B1 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    uint8_t                   EnterForceMode;                // 0x02B2 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    float                     ConstantForce;                 // 0x02B4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                     EnterForce;                  // 0x02B8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class AActor*             CustomForceDirection;        // 0x02C0 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ForceVolume_TA");
        }

        return uClassPointer;
    };

    void eventUnTouch(class AActor* Other);
    struct FVector GetForceDirection(class ARBActor_TA* RB);
    void eventTouch(class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector HitLocation, struct FVector HitNormal);
    void eventPrePhysicsStep(float DeltaTime);
};

// Class TAGame.FpsBucketRecorder_TA
// 0x001B (0x009D - 0x00B8)
class UFpsBucketRecorder_TA : public UActorComponent
{
public:
    TArray<struct FFpsBucketData>      Buckets;              // 0x00A0 (0x0010)
}

```

```

[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
int32_t MinFPS; // 0x00B0 (0x0004)
[0x0000000000000000]
int32_t MaxFPS; // 0x00B4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FpsBucketRecorder_TA");
}

return uClassPointer;
};

void eventConstruct();
struct FFpsBucketData GetTotalAverage();
TArray<struct FFpsBucketData> GetAverages();
};

// Class TAGame.FriendsListBuilder_TA
// 0x0030 (0x0060 - 0x0090)
class UFriendsListBuilder_TA : public UObject
{
public:
TArray<struct FPersonaDataId> PlatformFriends; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPersonaDataId> EpicFriends; // 0x0070 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPersonaDataId> AllFriends; // 0x0080 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FriendsListBuilder_TA");
}

return uClassPointer;
};

void Build(class UOnlineFriendMap_X* AllFriendsMap, class UOnlineFriendMap_X*
PlatformFriendsMap, class UOnlineFriendMap_X* EpicFriendsMap, class
ULinkedAccountMap_X* PlatformToEpicAccountMap, class UPersonas_TA* Personas);
};

```

```

// Class TAGame.FriendsListUtils_TA
// 0x0000 (0x0060 - 0x0060)
class UFriendsListUtils_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FriendsListUtils_TA");
}

return uClassPointer;
};

static struct FFriendInfoCounts UpdateFriendCounts(class UPersonas_TA* Personas,
TArray<struct FPersonaDataId>& Friends);
static TArray<struct FPersonaDataId> NetIdToPersonId(TArray<struct FUniqueNetId>& Ids);
static TArray<struct FUniqueNetId> PersonIdToNetId(TArray<struct FPersonaDataId>& Ids);
};

// Class TAGame.FXActor_GameEditorActor_TA
// 0x001C (0x0328 - 0x0344)
class AFXActor_GameEditorActor_TA : public AFXActor_X
{
public:
float SpeedMultiplier; // 0x0328 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UFXActorEvent_X* EditingState; // 0x0330 (0x0008)
[0x0000000000000000]
struct FVector PrevLocation; // 0x0338 (0x000C)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_GameEditorActor_TA");
}

return uClassPointer;
};

void HandleSelectionChange(class UInterface_GameEditor_TA* EditorObject, unsigned long bSelected);
void TickFX(float DeltaTime);
void PostBeginPlay();

```

```

};

// Class TAGame.FXActor_TA
// 0x00450 (0x0328 - 0x03678)
class AFXActor_TA : public AFXActor_X
{
public:
    class UFXActorEvent_X* [0x0000000000000000] HideWorldUIState; // 0x0328 (0x0008)
    class UFXActorEvent_X* [0x0000000000000000] PreviewState; // 0x0330 (0x0008)
    class UFXActorEvent_X* [0x0000000000000000] PaintedState; // 0x0338 (0x0008)
    class UFXActorEvent_X* [0x0000000000000000] BoostEndEvent; // 0x0340 (0x0008)
    class UFXActorEvent_X* [0x0000000000000000] StartOfReplayEvent; // 0x0348 (0x0008)
    TArray<class UFXActorEvent_X*> [0x000000000400000] (CPF_NeedCtorLink) TeamStates; // 0x0350 (0x0010)
    class UFXActorEvent_X* [0x0000000000000000] MyTeamState; // 0x0360 (0x0008)
    class UFXActorEvent_X* [0x0008] [0x0000000000000000] OpposingTeamState; // 0x0368
    class ULocalPlayerAudioParamsComponent_TA*
    LocalPlayerAudioParamsComponent; // 0x03670 (0x0008) [0x0000000004082008]
    (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FXActor_TA");
        }

        return uClassPointer;
    };

    static void __FXActor_TA__ClearPaintParameters_0x1(class UParticleSystemComponent* BeamPSC);
    void OnSetParticleSysParam(class USeqAct_SetParticleSysParam* Action);
    static void ClearPaintParameters(class AFXActor_X* FXActor);
    void ApplyPaintToAttachment(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, struct FFXAttachment& FXAttach);
    void ApplyPaint(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint);
    void HandleHideWorldUIChanged(class UGameShare_TA* G);
    void SetCarActorParam(class ACar_TA* InCar);
    void eventOnOwnerChanged();
    void PostBeginPlay();
};

```

```

// Class TAGame.FXActor_Ball_TA
// 0x0060 (0x03678 - 0x03CD8)
class AFXActor_Ball_TA : public AFXActor_TA
{
public:
    float SuperSonicSpeed; // 0x03678 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float MaxGoalProximityDistance; // 0x0367C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float BallFadeInSpeed; // 0x03780 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float BallFadeOutSpeed; // 0x03784 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float TrailOpacityLerpSpeed; // 0x03788 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float CurrentTrailOpacity; // 0x0378C (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float BallFadeSpeed; // 0x03890 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float BallFadeOpacity; // 0x03894 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float ShockwaveParamMin; // 0x03898 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float ShockwaveParamMax; // 0x0389C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float ShockwaveParamTime; // 0x039A0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UFXActorEvent_X* InAirState; // 0x039A8 (0x0008)
    [0x0000000000000000]
    class UFXActorEvent_X* SuperSonicState; // 0x03AB0 (0x0008)
    [0x0000000000000000]
    class ABall_TA* Ball; // 0x03AB8 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    float TouchingWorldTime; // 0x03BC0 (0x0004)
    [0x0000004000002000] (CPF_Transient)
    unsigned long bInAir : 1; // 0x03BC4 (0x0004)
    [0x0000004000002000] [0x00000001] (CPF_Transient)
    unsigned long bSuperSonic : 1; // 0x03BC4 (0x0004)
    [0x0000004000002000] [0x00000002] (CPF_Transient)
    float CurrentTrailDilation; // 0x03BC8 (0x0004)
    [0x0000004000002000] (CPF_Transient)
    float MaxTrailDilation; // 0x03BCC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FName TraillIntensityName; // 0x03CD0 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FXActor_Ball_TA");
        }
    }
}

```

```

}

return uClassPointer;
};

void __FXActor_Ball_TA__PostBeginPlay_0x2(class ABall_TA* _);
void __FXActor_Ball_TA__PostBeginPlay_0x1(class ABall_TA* _);
void eventDestroyed();
void UpdateBallPenetration();
void HandleBallExploded(class ABall_TA* InBall);
void HandleBallGameEventSet(class ABall_TA* InBall);
void ClearBallPositionMaterials();
void SetBallPositionMaterials(struct FLinearColor LocParam, float ShockwaveParam);
void UpdateBallPositionMaterials();
float GetClosestLocalPlayerDistance();
float GetPrimaryCameraDistance();
void UpdateTraillIntensity();
void UpdateTrailColor();
void UpdateGoalProximity();
void UpdateBallFade(float DeltaTime);
void StartBallFadeOut();
void StartBallFadeln();
void TickFX(float DeltaTime);
void PostBeginPlay();
};

// Class TAGame.FXActor_BallBreakout_TA
// 0x0060 (0x03CD8 - 0x04238)
class AFXActor_BallBreakout_TA : public AFXActor_Ball_TA
{
public:
    class UAkSoundCue* [0x0000000000000001] (CPF_Edit) MaxChargeSFX; // 0x03CD8 (0x0008)
    class UAkSoundCue* [0x0000000000000001] (CPF_Edit) SwitchTeamSFX; // 0x03DE0 (0x0008)
    class UAkSoundCue* [0x0000000000000001] (CPF_Edit) IdleSFX; // 0x03DE8 (0x0008)
    class UAkSoundCue* [0x0000000000000001] (CPF_Edit) ResetSFX; // 0x03EF0 (0x0008)
    TArray<class UAkSoundCue*> [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) ChargeSFX; // 0x03EF8 (0x0010)
    TArray<class UAkSoundCue*> [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) DamageSFX; // 0x03F408
    TArray<class UFXActorEvent_X*> [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) DamageEvents; // 0x04018
    class ABall_Breakout_TA* [0x000000000002000] (CPF_Transient) BallBreakout; // 0x04128 (0x0008)
    int32_t [0x0000000000000000] LastDamage; // 0x04230 (0x0004)
    float [0x0000000000000000] LastNoDamageHit; // 0x04234 (0x0004)

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_BallBreakout_TA");
}

return uClassPointer;
};

void UpdateGoalProximity();
void UpdateBallFade(float DeltaTime);
void StartBallFadeOut();
void StartBallFadeln();
void HandleAppliedDamage(class ABall_Breakout_TA* InBall, struct FAppliedBreakoutDamage
Damage);
void HandleLastTeamTouchChanged(class ABall_Breakout_TA* InBall, uint8_t
InLastTeamTouch);
void HandleDamageIndexChanged(class ABall_Breakout_TA* InBall, int32_t InDamageIndex);
void PostBeginPlay();
};

// Class TAGame.FXActor_BallHaunted_TA
// 0x0080 (0x03CD8 - 0x04458)
class AFXActor_BallHaunted_TA : public AFXActor_Ball_TA
{
public:
class UAkSoundCue*
[0x0000000000000001] (CPF_Edit)           SwitchTeamSFX;          // 0x03CD8 (0x0008)
class UAkSoundCue*
[0x0000000000000001] (CPF_Edit)           IdleSFX;              // 0x03DE0 (0x0008)
class UAkSoundCue*
[0x0000000000000001] (CPF_Edit)           BeamBreakSFX;         // 0x03DE8 (0x0008)
class UAkSoundCue*
[0x0000000000000001] (CPF_Edit)           BallEnterTrapSFX;     // 0x03EF0 (0x0008)
class UAkSoundCue*
[0x0000000000000001] (CPF_Edit)           BallExitTrapSFX;      // 0x03EF8 (0x0008)
class UAkSoundCue*
[0x0000000000000001] (CPF_Edit)           BallCountdownSFX;     // 0x03F400
(0x0008) [0x0000000000000001] (CPF_Edit)
TArray<class UFXActorEvent_X*>           DamageEvents;          // 0x03F408
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UParticleSystem*                    PulsePSArchetype;       // 0x04018 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                            PulsePSPParam;         // 0x04120 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                                 BallPulseTimeModifier;   // 0x04128 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                 BrokenBeamResetTime;    // 0x0412C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                 MaxTimePercentageToPerformPulse; // 0x04230 (0x0004)
[0x0000000000000001] (CPF_Edit)
class ABall_Haunted_TA*                 BallHaunted;          // 0x04238 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```

int32_t           DamageIndex;           // 0x04340 (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long      bBallExploded : 1;     // 0x04344 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long      bBeamIsBroken : 1;      // 0x04344 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
float              CurrentBrokenTime;    // 0x04348 (0x0004)
[0x0000000000002000] (CPF_Transient)
float              NextPulseTime;        // 0x0434C (0x0004)
[0x0000000000002000] (CPF_Transient)
class UParticleSystemComponent*   PulsePSC;    // 0x04450
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | 
CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_BallHaunted_TA");
}

return uClassPointer;
};

void __FXActor_BallHaunted_TA__PostBeginPlay_0x1(class ABall_Haunted_TA* _);
void UpdateGoalProximity();
void UpdateBallFade(float DeltaTime);
void SetTimeInGoalRTPC(float Value);
void ClearFXActor();
void HandleBallExploded(class ABall_TA* InBall);
void PulseBallInTrap();
void HandleIsTrapped(class ABall_Haunted_TA* InBall, unsigned long blsTrapped, class 
UGoal_TA* Goal);
void HandleLastTeamTouchChanged(class ABall_Haunted_TA* InBall, uint8_t InLastTeamTouch);
void SetBeamIsBroken(unsigned long bValue);
void SetDamageIndex(int32_t InDamageIndex);
void eventTick(float DeltaTime);
void StartBallFadeOut();
void StartBallFadeIn();
void HandleActiveRoundChanged(class AGameEvent_Soccar_TA* GameEvent);
void HandleGameEventSet(class ABall_TA* InBall);
void eventDestroyed();
void PostBeginPlay();
};

// Class TAGame.FXActor_Boost_TA
// 0x00C0 (0x03678 - 0x04238)
class AFXActor_Boost_TA : public AFXActor_TA
{
public:
TArray<struct FParticleParameterSet>      ParametersPerExtraBoostSockets;      //

```

```

0x03678 (0x0010) [0x000000000400000] (CPF_NeedCtorLink)
TArray<struct FParticleBodyParameterSet> BodyParticleParameters; // 
0x03788 (0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UAkSoundCue* DryFireSound; // 0x03898 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor CarBoostGlowColor; // 0x039A0 (0x0010)
[0x0000000000000001] (CPF_Edit)
TArray<struct FAnimatedMaterialMesh> BoostMeshMaterialParams; // 
0x03AB0 (0x0010) [0x000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
TArray<struct FBoostMeshMaterialOverride> BoostMeshMaterialOverrides; // 
0x03BC0 (0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UBoostMesh_TA* BoostMeshArchetype; // 0x03CD0
(0x0008) [0x0000000000000001] (CPF_Edit)
class UBoostMesh_TA* BoostMesh; // 0x03CD8 (0x0008)
[0x000000000002000] (CPF_Transient)
struct FRandomRange BoostMeshFadeInTime; // 0x03DE0
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FRandomRange BoostMeshFadeOutTime; // 0x03DE8
(0x0008) [0x0000000000000001] (CPF_Edit)
TArray<class UParticleSystemComponent*> MultipleBoostsOverride; // 
0x03EF0 (0x0010) [0x0000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF_NeedCtorLink | CPF>EditInline)
uint8_t BoostEmitterSocketBehavior; // 0x03F400 (0x0001)
[0x0000000000000001] (CPF_Edit)
TArray<struct FPreviewParticleOverride> PreviewParticleOverrides; // 
0x03F408 (0x0010) [0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
unsigned long bWasBoosting : 1; // 0x04018 (0x0004)
[0x000000000002000] [0x00000001] (CPF_Transient)
unsigned long bWasInputBoosting : 1; // 0x04018 (0x0004)
[0x000000000002000] [0x00000002] (CPF_Transient)
float MaxBoostMeshMaterialTime; // 0x0401C (0x0004)
[0x000000000002000] (CPF_Transient)
class AVehicle_TA* Vehicle; // 0x04120 (0x0008)
[0x000000000002000] (CPF_Transient)
class ACarPreviewActor_TA* CarPreview; // 0x04128 (0x0008)
[0x000000000002000] (CPF_Transient)
class UCarMeshComponentBase_TA* CarMesh; // 0x04230
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | 
CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_Boost_TA");
}

return uClassPointer;
};

void HandleUpdatedPaintSettings(class UCarMeshComponentBase_TA* CarMeshComponent);

```

```

void HandleCarDemolished(class ACar_TA* Car, struct FDemolishData Data);
void ApplyPaint(class UProductAttribute_PaintSettings_TA* PaintSettings, class
UProductPaint_TA* Paint);
void ApplyPreviewParticleOverrides();
void StopAllEffects();
void BoostMeshFadeOut();
void BoostMeshFadeIn();
void ResetBoostMeshElapsed Time(float Value);
void CopyBoostMeshDistributionParameters(class AFXActor_Boost_TA* Archetype);
void UpdateBoostMeshMaterials(float DeltaTime);
void InitializeBoostMesh();
void CreateBoostMeshMaterials(class AFXActor_X* FXActor);
void Inherit(class AFXActor_X* Other);
bool eventIsPRIInfoInGameState(struct FName GameState);
void TickFX(float DeltaTime);
void SetBodyParticleParams(class AFXActor_X* BodyFXActor);
void MoveToAverageBoostLocation(class UProductAsset_Body_TA* BodyAsset);
void InitExtraBoostComponents(class UProductAsset_Body_TA* BodyAsset);
void DuplicateAttachment(class UProductAsset_Body_TA* BodyAsset, int32_t AttachmentIdx);
void SetParticleParams(struct FName AttachmentName, TArray<struct FParticleSysParam>
Params);
void SetAttachmentActor(class AActor* AttachToActor);
};

// Class TAGame.FXActor_BreakoutPlatform_TA
// 0x0090 (0x03678 - 0x03F408)
class AFXActor_BreakoutPlatform_TA : public AFXActor_TA
{
public:
class UMaterialInterface* PlatformMaterial; // 0x03678 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UMaterialInterface* RaysMaterial; // 0x03780 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName LocationParam; // 0x03788 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FPlatformParam> DamagedParams; // 0x03890
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FPlatformParam> BrokenParams; // 0x039A0
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UAkSoundCue* DamageSFX; // 0x03AB0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue* DirectBreakSFX; // 0x03AB8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue* IndirectBreakSFX; // 0x03BC0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float DamageDistanceSpeed; // 0x03BC8 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UFXActorEvent_X* DamagedState; // 0x03CD0 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* BrokenState; // 0x03CD8 (0x0008)
[0x0000000000000000]
class ABreakOutActor_Platform_TA* BreakoutPlatform; // 0x03DE0
(0x0008) [0x0000000000002000] (CPF_Transient)
class UMaterialInstanceConstant* MIC; // 0x03DE8 (0x0008)

```

```

[0x0000000000000000] (CPF_Transient)          RaysMIC;           // 0x03EF0 (0x0008)
class UMaterialInstanceConstant*               [0x0000000000000000] (CPF_Transient)
[0x0000000000000000] (CPF_Transient)          uint8_t             DamageState;        // 0x03EF8 (0x0001)
float                                         DamageTime;         // 0x03EFC (0x0004)
[0x0000000000000000] (CPF_Transient)          float              BrokenTime;        // 0x03F400 (0x0004)
[0x0000000000000000] (CPF_Transient)          float              RemainingTime;     // 0x03F404 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_BreakoutPlatform_TA");
}

return uClassPointer;
};

void SetTeamColor(struct FLinearColor TeamColor);
void PlayDestroyIndirectSFX();
void PlayDestroyDirectSFX();
void PlayDamageSFX();
void UpdateParamsAtTime(TArray<struct FPlatformParam> Params, float Time);
void UpdateDamageParams(float DeltaTime);
void TickFX(float DeltaTime);
void HandleDamageStateChanged(class ABreakOutActor_Platform_TA* Platform, struct FBreakoutDamageState InState);
void SetAttachmentActor(class AActor* AttachToActor);
};

// Class TAGame.FXActor_Car_TA
// 0x011C (0x03678 - 0x04894)
class AFXActor_Car_TA : public AFXActor_TA
{
public:
class UEffectsMap_X*                      WheelEffectsMap;    // 0x03678 (0x0008)
[0x0000000000000001] (CPF_Edit)           class UAkSoundCue*   AkWheelImpactSound; // 0x03780
(0x0008) [0x0000000000000001] (CPF_Edit)  class UAkSoundCue*   AkWheelDriveSound; // 0x03788
(0x0008) [0x0000000000000001] (CPF_Edit)  class UAkSoundCue*   AkEnterSupersonicSound; // 0x03890
(0x0008) [0x0000000000000001] (CPF_Edit)  class UAkSoundCue*   AkLoopSupersonicSound; // 0x03898
(0x0008) [0x0000000000000001] (CPF_Edit)  float              MinImpactMomentum; // 0x039A0 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

struct FName AkImpactTypeKey; // 0x039A4 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName AkImpactIntensityKey; // 0x039AC (0x0008)
[0x0000000000000001] (CPF_Edit)
class UCameraShake* WheellImpactShake; // 0x03AB8
(0x0008) [0x0000000000000001] (CPF_Edit)
class UForceFeedbackWaveform* WheellImpactForceFeedback; // 0x03BC0 (0x0008) [0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat ShakeScaleCurve; // 0x03BC8 (0x0018)
[0x00000000040001] (CPF_Edit | CPF_NeedCtorLink)
class UParticleSystem* SupersonicWheelTemplate; // 0x03DE0
(0x0008) [0x0000000000000001] (CPF_Edit)
float BoostGlowRate; // 0x03DE8 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UShakeComponent_X* ShakeComponent; // 0x03EF0
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class AVehicle_TA* Vehicle; // 0x03EF8 (0x0008)
[0x000004000002000] (CPF_Transient)
unsigned long bThrottle : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bInAir : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bBrake : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bWasBoosting : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bWasBoostFlying : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bWasSuperSonic : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bWasOnWall : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bEnabled : 1; // 0x03F400 (0x0004)
[0x0000000000000001] (CPF_Transient)
unsigned long bPreferSimulatedState : 1; // 0x03F400 (0x0004)
[0x0001000000000001] (CPF_Transient)
class UFXActorEvent_X* ThrottleState; // 0x03F408 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* SuperSonicState; // 0x04010 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* InAirState; // 0x04018 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* BoostState; // 0x04120 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* BoostPreviewState; // 0x04128 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* BoostFlyState; // 0x04230 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* OnWallState; // 0x04238 (0x0008)
[0x0000000000000000]
class UFXActorEvent_X* SuperSonicEndState; // 0x04340
(0x0008) [0x0000000000000000]
class UFXActorEvent_X* PMCAAnimBeginState; // 0x04348

```

```

(0x0008) [0x0000000000000000] PMCAAnimEndState; // 0x04450
class UFXActorEvent_X* ContactPhysMat; // 0x04458 (0x0008)
(0x0008) [0x0000000000000000] class UPhysicalMaterial* ContactPhysMatProp; // 0x04560
[0x0000000000002000] (CPF_Transient) class UPhysicalMaterialProperty_TA* ContactPhysMatProp; // 0x04560
(0x0008) [0x0000000000002000] (CPF_Transient)
TArray<struct FWheelFX> WheelEffects; // 0x04568 (0x0010)
[0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink) class UParticleSystem* WheelEffectsParticle; // 0x04678 (0x0008)
[0x0000000000002000] (CPF_Transient) class UCarMeshComponent_TA* CarMesh; // 0x04780
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline) class UMaxActorsGroup_TA* MaxActorsGroup; // 0x04788
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float BoostGlow; // 0x04890 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_Car_TA");
}

return uClassPointer;
};

void PlayWheelLandCameraShake(float ImpactSpeed);
void SetWheelEffectActive(int32_t WheelIdx, unsigned long bEffectActive);
void SetContactPhysMat(class UPhysicalMaterial* NewPhysMat);
void UpdateContactPhysMat();
void QueryCarMovementState(struct FCarMovementState& OutState);
void PlaySpawnFX(class ACar_TA* Car);
void eventSetBraking(unsigned long bNewBraking);
void eventTickFX(float DeltaTime);
void HandleTeamChanged(class ACar_TA* InCar);
class AFXActor_TA* InstantiateTrailFXActor(class UProductAsset_SupersonicTrail_TA* TrailAsset, class AFXActor_TA* TrailFX, class ACar_TA* Car);
void InitSupersonicFX(class ACar_TA* Car, class UProductAsset_SupersonicTrail_TA* TrailAsset);
void HandleProductLoaded(class UProductAsset_TA* Product);
void HandleWheelLoaded(class UProductAsset_Wheel_TA* Product);
void HandleProductAssets(class UProductAsset_SupersonicTrail_TA* Product);
void InitWheelBones(class ACar_TA* Car);
void HandleVehicleSetup(class ACar_TA* Car);
void StopWheelEffects();
void StopAllEffects();
void HandleDisabled();

```

```

void HandleEnabled();
void eventDestroyed();
void PostBeginPlay();
};

// Class TAGame.FXTrait_BoostParticle_TA
// 0x004C (0x0060 - 0x00AC)
class UFXTrait_BoostParticle_TA : public UFXAttachmentTraitBase_X
{
public:
class UParticleSystem* [0x0000000000000001] (CPF_Edit) ParticlePreview; // 0x0060 (0x0008)
class UParticleSystem* [0x0000000000000001] (CPF_Edit) DuplicateParticle; // 0x0068 (0x0008)
class UParticleSystem* [0x0000000000000001] (CPF_Edit) DuplicateParticlePreview; // 0x0070 (0x0008)
TArray<struct FParticleSysParam> (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) SharedParameters; // 0x0078
TArray<struct FParticleSysParam> (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) DuplicateParameters; // 0x0088
TArray<struct FParticleSysParam> (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) PreviewOverrideParameters; // 0x0098
unsigned long bConvertedDuplicateParametersToSharedParameters : 1; // 0x00A8 (0x0004) [0x0000000000000002] [0x00000001] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXTrait_BoostParticle_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameData_TA
// 0x0258 (0x0060 - 0x02B8)
class UGameData_TA : public UObject
{
public:
class UProductDatabase_TA* [0x0000000000000003] (CPF_Edit | CPF_Const) Products; // 0x0060 (0x0008)
class UXPMManager_TA* [0x0000000000000001] (CPF_Edit) XPManger; // 0x0068 (0x0008)
int32_t [0x0000000000000003] (CPF_Edit | CPF_Const) MaxTeamSize; // 0x0070 (0x0004)
TArray<class UProductSlot_TA*> [0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink) ProductSlots; // 0x0078 (0x0010)
TArray<class UProductSlot_TA*> SlotLoadOrder; // 0x0088

```

(0x0010) [0x0000000000400003] (CPF\_Edit | CPF\_Const | CPF\_NeedCtorLink)  
class UProductSlot\_TA\* BodySlot; // 0x0098 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* SkinSlot; // 0x00A0 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* WheelSlot; // 0x00A8 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* BoostSlot; // 0x00B0 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* AntennaSlot; // 0x00B8 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* HatSlot; // 0x00C0 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* FrontSlot; // 0x00C8 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* PaintFinishSlot; // 0x00D0 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* BotSlot; // 0x00D8 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* LogoSlot; // 0x00E0 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* UnderGlowSlot; // 0x00E8 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* PremiumInventorySlot; // 0x00F0 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* CustomFinishSlot; // 0x00F8 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* EngineAudioSlot; // 0x0100 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* SupersonicTrailSlot; // 0x0108 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* GoalExplosionSlot; // 0x0110 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* PlayerBannerSlot; // 0x0118 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* PlayerTitleSlot; // 0x0120 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* ESportsTeamSlot; // 0x0128 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* ArchiveSlot; // 0x0130 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* AllSlot; // 0x0138 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* ShopItemSlot; // 0x0140 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* CurrencySlot; // 0x0148 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* PlayerAvatarSlot; // 0x0150 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* PlayerAvatarBorderSlot; // 0x0158 (0x0008)  
[0x0000000000000000]  
class UProductSlot\_TA\* GiftPackSlot; // 0x0160 (0x0008)  
[0x0001000000000000]  
class UProductSlot\_TA\* BlueprintSlot; // 0x0168 (0x0008)

```

[0x0001000000000000]
class UProductSlot_TA*           MusicStingersSlot;          // 0x0170 (0x0008)
[0x0000000000000000]
class FString                   AlwaysLoadedContentArchetype; // 0x0178 (0x0010)
[0x0000080000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
TArray<class FString>          TeamArchetypePaths;        // 0x0188 (0x0010)
[0x0000080000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
TArray<class ATeam_TA*>         TeamArchetypes;          // 0x0198 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
class FString                   GameModeListPath;         // 0x01A8 (0x0010)
[0x0000080000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
class UGameModeList_TA*          GameModeList;            // 0x01B8
(0x0008) [0x0000000000002002] (CPF_Const | CPF_Transient)
struct FLoadoutData             DefaultLoadouts[0x2];    // 0x01C0 (0x0080)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FClientLoadoutData       DefaultClientLoadout;     // 0x0240 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class UCarColorSet_TA*          CustomColorSet;          // 0x0250 (0x0008)
[0x0000000000000000]
class UCarColorSet_TA*          ClubColorSet;            // 0x0258 (0x0008)
[0x0000000000000000]
class UCarColorSet_TA*          ClubColorSet_v1;         // 0x0260 (0x0008)
[0x0000000000000000]
class UCarColorSet_TA*          PlayerBannerColorSet;    // 0x0268 (0x0008)
[0x0000000000000000]
class UCarColorSet_TA*          PlayerVanityColorSet;    // 0x0270 (0x0008)
[0x0000000000000000]
class UMapList_TA*              MapList;                 // 0x0278 (0x0008)
[0x0000000000000001] (CPF>Edit)
class UGlobalLookupTexture_TA*   GlobalLookupTexture;      // 0x0280
(0x0008) [0x0000000000000000]
TArray<class UClass*>           OnlineProductAttributes; // 0x0288 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class UPaintDatabase_TA*         PaintDB;                // 0x0298 (0x0008)
[0x0000000000000003] (CPF>Edit | CPF_Const)
class UCertifiedStatDatabase_TA* CertifiedStatDB;        // 0x02A0
(0x0008) [0x0000000000000003] (CPF>Edit | CPF_Const)
class USpecialEditionDatabase_TA* SpecialEditionDB;       // 0x02A8
(0x0008) [0x0000000000000003] (CPF>Edit | CPF_Const)
class UDataAssetDatabase_ESportsTeam_TA*   ESportsTeamDB;    //
0x02B0 (0x0008) [0x0000000000000003] (CPF>Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameData_TA");
}

return uClassPointer;
};

```

```

static class UCarColorSet_TA* GetClubColorSet();
static class FString GetMapName(unsigned long bLocalize);
static TArray<class UProductSlot_TA*> GetUnlockableLoadoutSlots();
static TArray<class UProductSlot_TA*> GetVanityLoadoutSlots();
static TArray<class UProductSlot_TA*> eventGetLoadoutSlots();
static TArray<class UProductSlot_TA*> GetGarageLoadoutSlots();
static bool CanProductBelnSlot(class UProduct_TA* Product, class UProductSlot_TA* Slot);
static class UProductDatabase_TA* GetProductDB();
static void SetDefaultTeamColors();
static void eventSetGlobalTeamColors(TArray<struct FLinearColor> Colors);
};

// Class TAGame.GameEngine_TA
// 0x0008 (0x0B48 - 0x0B50)
class UGameEngine_TA : public UGameEngine_X
{
public:
    class UServerConfig_X*           ServerConfig;          // 0x0B48 (0x0008)
    [0x0000800000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GameEngine_TA");
        }

        return uClassPointer;
    }

    bool __GameEngine_TA__FlushOnlineSaveData_0x3(class UEOSOnlineStorageUploader_TA* U);
    bool __GameEngine_TA__FlushOnlineSaveData_0x2(class UEOSOnlineStorageUploader_TA* U);
    class UEOSOnlineStorageUploader_TA* __GameEngine_TA__FlushOnlineSaveData_0x1(class
        ULocalPlayer_TA* P);
    void FlushOnlineSaveData();
    static class FString GetBuildInfo(class FString Key);
    void eventConstruct();
};

// Class TAGame.GameEvent_TA
// 0x04D0 (0x0268 - 0x0738)
class AGameEvent_TA : public AActor
{
public:
    class UGroupComponent_ORS*         RegistryGroup;        // 0x0268
    (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component |
    CPF_EditInline)
    class UMatchType_TA*              MatchType;           // 0x0270 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UClass*                     MatchTypeClass;       // 0x0278 (0x0008)
};

```

```

[0x0000000100000021] (CPF_Edit | CPF_Net)
struct FName EventType; // 0x0280 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FName AudioSoundMode; // 0x0288 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName MatchTypeName; // 0x0290 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class ACar_TA* CarArchetype; // 0x0298 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t CountdownTime; // 0x02A0 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t FinishTime; // 0x02A4 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bMultiplayer : 1; // 0x02A8 (0x0004)
[0x0000000000000003] [0x00000001] (CPF_Edit | CPF_Const)
unsigned long bCountdownMessagesDisabled : 1; // 0x02A8
(0x0004) [0x0000000000000000] [0x00000002]
unsigned long bAllowPrespawn : 1; // 0x02A8 (0x0004)
[0x0001000000000000] [0x00000004]
unsigned long bFillWithAI : 1; // 0x02A8 (0x0004)
[0x0000000000000001] [0x000000048] (CPF_Edit)
unsigned long bAllowQueueSaveReplay : 1; // 0x02A8 (0x0004)
[0x0000000000000001] [0x000000108] (CPF_Edit)
unsigned long bAllowReadyUp : 1; // 0x02A8 (0x0004)
[0x0000004000002020] [0x000000120] (CPF_Net | CPF_Transient)
unsigned long bRestartingMatch : 1; // 0x02A8 (0x0004)
[0x0000000000000000] [0x000000240]
unsigned long bHasLeaveMatchPenalty : 1; // 0x02A8 (0x0004)
[0x0000004100002020] [0x000000480] (CPF_Net | CPF_Transient)
unsigned long bCanVoteToForfeit : 1; // 0x02A8 (0x0004)
[0x0000004100002020] [0x000001080] (CPF_Net | CPF_Transient)
unsigned long bAwardAchievements : 1; // 0x02A8 (0x0004)
[0x0000000000000000] [0x000001200]
unsigned long bIsBotMatch : 1; // 0x02A8 (0x0004)
[0x0000004000000020] [0x000002400] (CPF_Net)
unsigned long bAllowTeamVoiceChat : 1; // 0x02A8 (0x0004)
[0x0001008000000000] [0x000004800]
unsigned long bAllowSpectatorVoiceChat : 1; // 0x02A8 (0x0004)
[0x0001008000000000] [0x000010800]
int32_t MinPlayers; // 0x02AC (0x0004)
[0x0000004000000001] (CPF_Edit)
int32_t MaxPlayers; // 0x02B0 (0x0004)
[0x0000004000000001] (CPF_Edit)
TArray<class AActor*> SpawnPoints; // 0x02B8 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class AAIManager_TA* AIManagerArchetype; // 0x02C8
(0x0008) [0x0000000000000001] (CPF_Edit)
class AAIManager_TA* AIManager; // 0x02D0 (0x0008)
[0x0000004000002000] (CPF_Transient)
class APylon_Soccar_TA* Pylon; // 0x02D8 (0x0008)
[0x0008000000000001] (CPF_Edit)
class AAIController_TA* BotArchetype; // 0x02E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float BotSkill; // 0x02E8 (0x0004)

```

```

[0x0000004000002020] (CPF_Net | CPF_Transient)
TArray<class UMutator_TA*> DefaultMutators; // 0x02F0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UMutator_TA*> Mutators; // 0x0300 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
float MatchTimeDilation; // 0x0310 (0x0004)
[0x0000004000000001] (CPF_Edit)
class AHUD* HUDArchetype; // 0x0318 (0x0008)
[0x0002000000000001] (CPF_Edit)
class APlayerController_TA* Activator; // 0x0320 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ACar_TA* ActivatorCar; // 0x0328 (0x0008)
[0x0000004000002020] (CPF_Net | CPF_Transient)
TArray<class AController*> Players; // 0x0330 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class APRI_TA*> PRIs; // 0x0340 (0x0010)
[0x00000048000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class ACar_TA*> Cars; // 0x0350 (0x0010)
[0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class APlayerController_TA*> LocalPlayers; // 0x0360
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t StartPointIndex; // 0x0370 (0x0004)
[0x0000008000002000] (CPF_Transient)
uint8_t ReplicatedStateIndex; // 0x0374 (0x0001)
[0x0000000100002022] (CPF_Const | CPF_Net | CPF_Transient)
TArray<struct FName> DeprecatedIndexToState; // 0x0378
(0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FName ReplicatedStateName; // 0x0388 (0x0008)
[0x0000000100002020] (CPF_Net | CPF_Transient)
class UMessage_TA* GameFullMessage; // 0x0390 (0x0008)
[0x0000000000000000]
class UMessage_TA* OtherActiveMessage; // 0x0398 (0x0008)
[0x0000000000000000]
class UMessage_TA* NotEnoughPlayersMessage; // 0x03A0
(0x0008) [0x0000000000000000]
class UMessage_TA* TooManyPlayersMessage; // 0x03A8
(0x0008) [0x0000000000000000]
class UMessage_TA* NotPartyLeaderMessage; // 0x03B0
(0x0008) [0x0000000000000000]
TArray<class UMessage_TA*> CountdownMessages; // 0x03B8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UMessage_TA* GoMessage; // 0x03C8 (0x0008)
[0x0000000000000000]
TArray<class UMessage_TA*> ReplacingBotCountdownMessages; // 
0x03D0 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UMessage_TA* ReplacingBotGoMessage; // 0x03E0
(0x0008) [0x0000000000000000]
class USpawnDelayConfig_TA* SpawnDelayConfig; // 0x03E8
(0x0008) [0x0000800000000000]
TArray<struct FPlayerChatMessage> ChatHistory; // 0x03F0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t GameStateTimeRemaining; // 0x0400 (0x0004)
[0x0000004000002000] (CPF_Transient)
int32_t ReplicatedGameStateTimeRemaining; // 0x0404 (0x0004)

```

```

[0x0000000100002020] (CPF_Net | CPF_Transient)
class UBotConfig_TA*           BotConfig;          // 0x0408 (0x0008)
[0x0000800000000000]
TArray<struct FUniqueNetId>   ForfeitInitiatorIDs; // 0x0410 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FUniqueNetId>   BannedPlayers;      // 0x0420 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class APRI_TA*                GameOwner;         // 0x0430 (0x0008)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FCustomMatchSettings   MatchSettings;     // 0x0438 (0x0090)
[0x0000004100402020] (CPF_Net | CPF_Transient | CPF_NeedCtorLink)
class FString                 RichPresenceString; // 0x04C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UMatchAdminComponent_TA* MatchAdmin;        // 0x04D8
(0x0008) [0x0000000000002000] (CPF_Transient)
class UListenServer_TA*       ListenServer;      // 0x04E0 (0x0008)
[0x0000004000000000]
int32_t                      ReplicatedRoundCountDownNumber; // 0x04E8 (0x0004)
[0x0000004100000020] (CPF_Net)
class UMatchRecorder_TA*      MatchRecorder;     // 0x04F0 (0x0008)
[0x0000004000000000]
TArray<class UProductAsset_Bot_TA*> RandomizedBots; // 0x04F8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<class UGameState_TA*>   GameStates;        // 0x0508
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UGameState_TA*          GameState;         // 0x0518 (0x0008)
[0x0000004000000000]
class UPauseStaticDataSync_X* PauseInstance;      // 0x0520
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate         __EventMatchStarted__Delegate; // 0x0528
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventGameStateChanged__Delegate; // 0x0540
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventGameStateTimeUpdated__Delegate; // 0x0558
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventFinished__Delegate;    // 0x0570
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventDestroyed__Delegate; // 0x0588
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventPlayerRestarted__Delegate; // 0x05A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventPlayerAdded__Delegate; // 0x05B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventPlayerRemoved__Delegate; // 0x05D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventCarAdded__Delegate;   // 0x05E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventCarRemoved__Delegate; // 0x0600
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventReplicatedDemolish__Delegate; // 0x0618
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventPenaltyChanged__Delegate; // 0x0630
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventCanVoteForfeitChanged__Delegate; // 0x0648

```

```

(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCountDownTimeUpdated__Delegate;      // 0x0660
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventGameOwnerUpdated__Delegate;        // 0x0678
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventMatchTypeSet__Delegate;         // 0x0690
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventLocalPlayersChanged__Delegate;   // 0x06A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventMatchSettingsChanged__Delegate; // 0x06C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPlayerResetTraining__Delegate; // 0x06D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventGameEventCommand__Delegate;     // 0x06F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventRestartPlayer__Delegate;       // 0x0708
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __Pylon__ChangeNotify;                // 0x0720 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_TA");
}

return uClassPointer;
};

void InitCountDown();
void StartCountdownTimer();
void AllowReadyUp();
class APRI_TA* FindPlayerPRI(struct FUniqueNetId UniqueId);
void HandlePlayerRemoved(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void UpdateGameOwner();
void SetGameOwner(class APRI_TA* NewOwner);
void __GameEvent_TA__PostBeginPlay_0x1(class UOnlineGameDedicatedServer_X* _, struct FCustomMatchSettings Settings);
class UMutator_TA* __GameEvent_TA__GetOnlineGameSettingsMutators_0x1(class UMutatorGameSetting_TA* Setting);
void __GameEvent_TA__InitMutators_0x1(class UMutator_TA* Mutator);
void __GameEvent_TA__SetAllowReadyUp_0x1(class APRI_TA* P);
bool __GameEvent_TA__CheckPlayersReady_0x1(class APRI_TA* P);
struct FUniqueNetId __GameEvent_TA__CheckForBannedPlayers_0x2(class APRI_TA* PRI);
bool __GameEvent_TA__CheckForBannedPlayers_0x1(class APRI_TA* PRI);
void __Pylon__ChangeNotifyFunc();
bool ShouldClipCameraToField();
void AddAdditionalViewableActors(TArray<class FString>& Actors);
bool DisableNamePlates();
bool GetPlayerHUDPosition(class APRI_TA* PRI, struct FVector& Out_Position);

```

```
bool AllowPausing(class APlayerController* PC);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void GameEventCommand(class APlayerController* Caller, class FString CommandType);
void PlayerResetTraining();
bool SuppressModalDialogs();
bool ShouldShowBallIndicator();
class UGameMode_TA* GetGameMode();
void CheckInitiatedForfeit(class APRI_TA* PRI);
void HandleSyncedBanStatus(class UBansync_TA* BanSync);
class APlayerController_TA* FindPCForUniqueId(struct FUniqueNetId PlayerID);
void CheckForBannedPlayers();
bool AllowSplitScreenPlayer();
void AddPlayerChatMessage(struct FUniqueNetId PlayerID, uint8_t ChatChannel, class ATeamInfo* Team, class FString Message);
void ConditionalStartSpectatorMatch();
static class FString LocalizePresence(class FString GameDataString);
void GetPresenceStatus(unsigned long bLocalize, class FString& Out_PresenceString, class FString& Out_GameDataString);
bool ShouldShowDisconnectedPlayersOnScoreboard();
int32_t ScoreboardSort(class APRI_TA* A, class APRI_TA* B);
struct FName GetMatchTypeName();
bool IsPlayingFTE();
bool IsPlayingTraining();
bool IsPlayingLan();
bool IsPlayingOffline();
bool IsPlayingPrivate();
bool IsPlayingPublic();
bool IsOnlineMultiplayer();
void CreateMatchType(class FString Options);
bool AllPlayersSelectedTeam();
int32_t PlayersOnFieldCount();
int32_t PlayersSelectedTeamCount(int32_t& PlayersHavingSelected, int32_t& TotalPlayers, int32_t& Spectators);
bool CanQueSaveReplay();
void ForceMatchStart();
void ConditionalStartMatch();
bool DisableStatXP();
bool GiveExtraXP();
void SaveLocalPlayerStats();
bool CanUseBallCam();
bool HandleNextGame();
struct FName GetDesiredSoundGameMode();
struct FName GetDesiredSoundState();
void SetMaxPlayers(int32_t InMaxPlayers);
void SetRestartingMatch(unsigned long bRestart);
class UIOnlineGameHost_X* GetOnlineGameHost();
class UOnlineGameDedicatedServer_TA* GetOnlineGameDedicatedServer();
bool ShouldBeFullScreen();
bool IsPostMatch();
bool IsFinished();
void OnAllPlayersReady();
void CheckPlayersReady();
void SetAllowReadyUp(unsigned long bAllow);
void AutoReadyPlayers();
```

```
bool ShouldAutoReadyUp(class APRI_TA* PRI);
void SendGoMessage(class APlayerController_TA* Player);
void SendCountdownMessage(int32_t Seconds, class APlayerController_TA* Player);
void BroadcastCountdownMessage(int32_t Seconds);
void BroadcastGoMessage();
bool AllowShutdown();
void HandleCarAddCarComponent(class UObject* O);
float GetRealDeltaTime(float ElapsedTime);
void SetTimeDilation(float NewTimeDilation);
void ReplaceBotsWithAwaitingPlayers();
void OnReplicatedDemolish(struct FDemolishData Data);
void RemoveCar(class ACar_TA* Car);
void AddCar(class ACar_TA* Car);
void SetBotSkillByTeam(int32_t Team, float NewSkill, float BoostThreshold);
void SetBotSkill(float NewSkill);
class AAIController_TA* SpawnBotController();
class APlayerController_TA* GetLocalPrimaryPlayer();
void AddBot(class AAIController_TA* Bot, class UProductAsset_Bot_TA* BotAsset);
bool GetGeneratedBotName(class FString& Out_BotName);
class AAIController_TA* SpawnBot();
class UProductAsset_Bot_TA* ChooseBotAsset(class AAIController_TA* Bot);
bool HasPlayerNamed(class FString PlayerName);
void RandomizeBots();
bool MoveToGround(class AActor* Mover, float HeightCheck);
void SetAllDriving(unsigned long bDriving);
void eventDestroyed();
void OnFinished();
void StartEvent();
void StartCountDown();
void StartInitialCountDown();
void OnGameStateTimeLapsed();
void OnGameStateTimeUpdated();
void UpdateGameStateTime();
void SetGameStateTimeRemaining(int32_t StateTime, unsigned long bFromReplication);
void SetGameStateTime(int32_t StateTime);
class UGameState_TA* GetGameState(struct FName StateName);
void SetGameStateComponents(struct FName StateName);
void GotoStateInternal(struct FName StateName);
void GotoGameState(struct FName StateName);
void OnPlayerRestarted(class ACar_TA* PlayerCar);
void TeleportCar(class ACar_TA* PlayerCar);
void OnCarSpawned(class ACar_TA* NewCar);
class ACar_TA* SpawnCar(class AController* NewPlayer, struct FVector SpawnLoc, struct FRotator SpawnRot);
bool FindValidCarSpawnOffset(struct FRotator StartRotation, int32_t TestCount, struct FVector& out_Location);
bool SpotIsEncroached(struct FVector Spot);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct FRotator& out_Rotation);
bool CanSpectate();
void RandomizeSpawnPoints();
bool RestartPlayer(class AController* NewPlayer);
bool IsPlayerSplitscreenChild(class UPlayer* InPlayer);
bool ShouldDelaySplitscreenPlayerRestart(class UPlayer* InPlayer);
```

```
bool PlayerCanRestart(class AController* aPlayer);
void RestartPlayers();
void RemoveLocalPlayer(class APlayerController_TA* Player);
void AddLocalPlayer(class APlayerController_TA* Player);
void RemovePRI(class APRI_TA* PRI);
void AddPRI(class APRI_TA* PRI);
void AddForfeitInitiator(struct FUniqueNetId PlayerID);
void BanPlayerID(struct FUniqueNetId PlayerID);
void BanConnection(class UNetConnection* Connection);
void OnPlayerRemoved(class AController* OldPlayer);
void RemovePlayer(class AController* Player);
int32_t GetMaxHumans();
int32_t GetNumHumans();
void AddPlayers(TArray<class AController*> PlayersToAdd);
bool ReplacePlayer(class APRI_TA* NewPlayer, struct FVector& CarLocation, struct FRotator& CarRotation, struct FVector& CarVelocity);
bool ReplaceBot(class APRI_TA* PRI, class APRI_TA* Replace, struct FVector& CarLocation, struct FRotator& CarRotation, struct FVector& CarVelocity);
bool FindBotReplacement(class APRI_TA* PRI);
void UpdateBotCount();
void TimerUpdateBotCount();
void TryRestartPlayer(class APRI_TA* PlayerPRI, struct FScriptDelegate RestartPlayerCallback);
void HandleSelectedLoadout(class APRI_TA* PlayerPRI);
void OnPlayerAdded(class AController* NewPlayer);
bool AddPlayer(class AController* Player, class UMessage_TA*& ErrorMsg);
void ResetPickups();
void ResetPlayers();
void DestroyCars();
bool IsAIDebugger(class AController* C);
void OnInit();
void HandleAverageSkillChanged(class UOnlineGameDedicatedServer_X* Server);
void InitBotSkill();
void MutateObject(class UObject* O);
bool HasMutatorType(class UClass* MutatorClass);
bool HasMutatorNamed(struct FName MutatorName);
void InitMutators();
TArray<class UMutator_TA*> GetOnlineGameSettingsMutators();
void HandleDestroyed(class AGameEvent_TA* GameEvent);
void Init(class APlayerController_TA* InActivator);
class UClass* GetMatchTypeFromOptions(class FString Options);
void InitListenServer();
void eventInitGame(class FString Options);
bool PlayerCanStartEvent(class APlayerController_TA* InActivator, class UMessage_TA*& ErrorMsg);
class UOnlineGame_Base_X* GetOnlineGame();
void NotifyKismetGameStateChanged();
void OnGameStateChanged();
void OnCanVoteForfeitChanged();
void UpdateCanVoteToForfeit();
bool ShouldAllowVoteToForfeit();
void OnPenaltyChanged();
void UpdateLeaveMatchPenalty();
bool IsPlaylistRanked();
class UGameSettingPlaylist_X* GetPlaylist();
```

```

bool ShouldHaveLeaveMatchPenalty();
class UGameMetrics_TA* GetMetrics();
void OnMatchSettingsChanged();
void SetCustomMatchSettings(struct FCustomMatchSettings NewSettings);
void ClearGameScoreFromCustomSettings();
void eventReplicatedEvent(struct FName VarName);
class FString MatchGuid();
void HandleConnectionTimedOut(class APlayerController* PC);
void eventPostBeginPlay();
bool EventRestartPlayer(class AController* NewPlayer);
void EventGameEventCommand(class APlayerController* Caller, class FString CommandType);
void EventPlayerResetTraining(class AGameEvent_TA* GameEvent);
void EventMatchSettingsChanged(class AGameEvent_TA* GameEvent, struct FCustomMatchSettings NewMatchSettings);
void EventLocalPlayersChanged(class AGameEvent_TA* GameEvent);
void EventMatchTypeSet(class AGameEvent_TA* GameEvent);
void EventGameOwnerUpdated(class AGameEvent_TA* GameEvent, class APRI_TA* NewGameOwner);
void EventCountDownTimeUpdated(class AGameEvent_TA* GameEvent, int32_t Seconds);
void EventCanVoteForfeitChanged(class AGameEvent_TA* GameEvent);
void EventPenaltyChanged(class AGameEvent_TA* GameEvent);
void EventReplicatedDemolish(class AGameEvent_TA* GameEvent, struct FDemolishData Data);
void EventCarRemoved(class AGameEvent_TA* GameEvent, class ACar_TA* Car);
void EventCarAdded(class AGameEvent_TA* GameEvent, class ACar_TA* Car);
void EventPlayerRemoved(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void EventPlayerAdded(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void EventPlayerRestarted(class AGameEvent_TA* GameEvent, class ACar_TA* PlayerCar);
void EventDestroyed(class AGameEvent_TA* GameEvent);
void EventFinished(class AGameEvent_TA* GameEvent);
void EventGameStateTimeUpdated(class AGameEvent_TA* GameEvent);
void EventGameStateChanged(class AGameEvent_TA* GameEvent);
void EventMatchStarted(class AGameEvent_TA* GameEvent);
};


```

```

// Class TAGame.GameEvent_Team_TA
// 0x0068 (0x0738 - 0x07A0)
class AGameEvent_Team_TA : public AGameEvent_TA
{
public:
TArray<class ATTeam_TA*> TeamArchetypes; // 0x0738 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class ATTeam_TA*> Teams; // 0x0748 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t MaxTeamSize; // 0x0758 (0x0004)
[0x0000008000002020] (CPF_Net | CPF_Transient)
int32_t NumBots; // 0x075C (0x0004)
[0x0000008000002000] (CPF_Transient)
unsigned long bMuteOppositeTeams : 1; // 0x0760 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bDisableMutingOtherTeam : 1; // 0x0760 (0x0004)
[0x0000004000002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long bForfeit : 1; // 0x0760 (0x0004)
[0x0000004000002020] [0x00000004] (CPF_Net | CPF_Transient)
unsigned long bUnfairTeams : 1; // 0x0760 (0x0004)

```

```
[0x0000004000002000] [0x00000008] (CPF_Transient)
unsigned long bAlwaysAutoSelectTeam : 1; // 0x0760 (0x0004)
[0x0000000000000000] [0x00000010]
class AVoteActor_TA* RematchVoteArchetype; // 0x0768
(0x0008) [0x0000000000000001] (CPF_Edit)
class AVoteActor_TA* RematchVote; // 0x0770 (0x0008)
[0x0000004000002000] (CPF_Transient)
TArray<class UProductAsset_TA*> PreloadedBotAssets; // 0x0778
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __EventTeamsCreated__Delegate; // 0x0788
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Team_TA");
}

return uClassPointer;
};

void StartRematchVote();
void CheckRematchVote();
bool __GameEvent_Team_TA__AllTeamsHaveHumans_0x1(class ATeam_TA* Team);
bool SplitScoreboardByTeams();
void UpdatePlayerShortcuts();
void ClearTemporarySpawnSpots();
void ResetPlayers();
class ATeam_TA* GetOtherTeam(class ATeamInfo* Team);
bool ChooseTeam(int32_t TeamIndex, class APlayerController_TA* Player);
void OnPenaltyChanged();
void OnTeamForfeited(class ATeam_TA* Team);
void HandleTeamForfeit(class ATeam_TA* Team);
void MuteOppositeTeams(unsigned long bMute);
void UpdateBotCount();
bool ReplacePlayer(class APRI_TA* PRI, struct FVector& CarLocation, struct FRotator& CarRotation, struct FVector& CarVelocity);
bool ReplaceBot(class APRI_TA* PRI, class APRI_TA* Replace, struct FVector& CarLocation, struct FRotator& CarRotation, struct FVector& CarVelocity);
bool FindBotReplacement(class APRI_TA* PRI);
void EndGame();
void ForceNoContest();
void CheckForForfeit(class AController* RemovedPlayer, class ATeam_TA* Team);
void OnPlayerRemoved(class AController* OldPlayer);
class ATeam_TA* PickTeam(class AController* C);
bool ShouldAutoSelectTeam(class AController* NewPlayer);
void AddTemporarySpawnSpot(class ATeam_TA* Team, class ACar_TA* Car);
bool RestartPlayer(class AController* NewPlayer);
void SetPlayerTeam(class AController* Player, int32_t Team);
void RemovePlayerFromTeam(class AController* Player);
```

```

void AddPlayerToTeam(class ATeam_TA* NewTeam, class AController* NewPlayer);
void SelectPlayerTeam(class APRI_TA* PRI);
void HandleSelectedLoadout(class APRI_TA* PlayerPRI);
void OnPlayerAdded(class AController* NewPlayer);
void eventDestroyed();
void DestroyTeams();
void NotifyAllTeamsCreated(struct FScriptDelegate Callback);
void OnAllTeamsCreated();
void AssignCustomTeamSettings();
void OnMatchSettingsChanged();
bool AllTeamsHaveHumans();
bool AllTeamsCreated();
void SetTeam(int32_t TeamNum, class ATeam_TA* NewTeam);
void CreateTeams();
void PreloadBot(class UProductAsset_Bot_TA* Bot);
void PreloadBots();
void OnInit();
void SetMaxTeamSize(int32_t MaxSize);
void UpdateMaxTeamSize();
void SetUnfairTeams(unsigned long bUnfair);
void InitBotSkill();
void eventInitGame(class FString Options);
void EventTeamsCreated(class AGameEvent_Team_TA* GameEvent);
};

// Class TAGame.GameMode_TA
// 0x005860 (0x0060 - 0x00B8C0)
class UGameMode_TA : public UObject
{
public:
class AGameInfo* [0x0000000800000001] (CPF_Edit) GameArchetype; // 0x0060 (0x0008)
class UMapSet_TA* [0x0000000000000001] (CPF_Edit) MapSet; // 0x0068 (0x0008)
class UMapSet_TA* [0x0000000000000001] (CPF_Edit) FreeplayMapSet; // 0x0070 (0x0008)
class UMapSet_TA* [0x0000000000000001] (CPF_Edit) StandardMapSet; // 0x0078 (0x0008)
class UMapSet_TA* [0x0000000000000001] (CPF_Edit) RocketLabsMedleyMapSet; // 0x0080
struct FName [0x0008] [0x0000000000000001] (CPF_Edit) MutatorPresetName; // 0x00808 (0x0008)
TArray<struct FName> [0x0010] [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) DisallowedMutatorCategories; // 0x008890
unsigned long [0x0000000000000000] [0x00000001] bEnabled : 1; // 0x0098A0 (0x0004)
unsigned long [0x0000000000000000] [0x00000002] bAllowPrivateMatchOnly : 1; // 0x00A0 (0x0004)
class UClass* [0x0000000000000000] GameInfoClass; // 0x00A08 (0x0008)
class FString [0x0000000000400000] (CPF_NeedCtorLink) GameArchetypePath; // 0x00A8B0 (0x0010)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameMode_TA");
}

return uClassPointer;
};

class FString GetLocalizedDescription();
class FString GetLocalizedName();
};

// Class TAGame.GameModeList_TA
// 0x0010 (0x0060 - 0x0070)
class UGameModeList_TA : public UObject
{
public:
TArray<class UGameMode_TA*> GameModes; // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameModeList_TA");
}

return uClassPointer;
};

class UGameMode_TA* GetGameModeForFreeplayMap(struct FName MapName);
class UGameMode_TA* GetGameModeForClass(class UClass* GameInfoClass);
class UMapSet_TA* GetRocketLabsMedleyMapSet(int32_t GameModelIndex);
class UMapSet_TA* GetStandardMapSet(int32_t GameModelIndex);
class UMapSet_TA* GetMapSet(int32_t GameModelIndex);
class FString GetGameInfoClass(int32_t GameModelIndex);
class FString GetGameURL(int32_t GameModelIndex);
};

// Class TAGame.GameObserver_TA
// 0x0194 (0x00A4 - 0x0238)
class UGameObserver_TA : public UActorComponent_X
{
public:
float HitTimeToBallMin; // 0x00A8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float HitTimeToBallMax; // 0x00AC (0x0004)
};

```

```
[0x0000000000000001] (CPF_Edit)
float HitPredictDistMax; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float HitPredictDistMin; // 0x00B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float OffenseDotBetweenMin; // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float OffenseDotBetweenMax; // 0x00BC (0x0004)
[0x0000000000000001] (CPF_Edit)
float OffenseDistToGoalMin; // 0x00C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float OffenseDistToGoalMax; // 0x00C4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float OffenseDistToBallMin; // 0x00C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float OffenseDistToBallMax; // 0x00CC (0x0004)
[0x0000000000000001] (CPF_Edit)
float DefenseDotBetweenMin; // 0x00D0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DefenseDotBetweenMax; // 0x00D4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DefenseDistToGoalMin; // 0x00D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DefenseDistToGoalMax; // 0x00DC (0x0004)
[0x0000000000000001] (CPF_Edit)
float DefenseDistToBallMin; // 0x00E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float DefenseDistToBallMax; // 0x00E4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDistToBallMin; // 0x00E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDistToBallMax; // 0x00EC (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDistToBallWeight; // 0x00F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDotToBallMin; // 0x00F4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDotToBallMax; // 0x00F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDotToBallWeight; // 0x00FC (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityOffenseWeight; // 0x0100 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ScorabilityDefenseWeight; // 0x0104 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarBallAerialBlend; // 0x0108 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxAerialHitProbability; // 0x010C (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bDebugCarData : 1; // 0x0110 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long bDebugGoalData : 1; // 0x0110 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long bDebugBallData : 1; // 0x0110 (0x0004)
```

```

[0x0000000000000000] [0x00000004]
unsigned long          bDebugPrediction : 1;           // 0x0110 (0x0004)
[0x0000000000000000] [0x00000008]
TArray<struct FCarData>      RelevantCars;           // 0x0118 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FGoalData>      RelevantGoals;          // 0x0128 (0x0010)
[0x0000000000480000] (CPF_Component | CPF_NeedCtorLink)
float                      MinAerialAltitude;        // 0x0138 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      MaxAerialAltitude;        // 0x013C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                      AerialLikelyhood;         // 0x0140 (0x0004)
[0x0000004000000000]
float                      MaxCalculatedTimeToBall;   // 0x0144 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FBallData           RelevantBall;            // 0x0148 (0x0020)
[0x0000000000000000]
TArray<class UObject*>      Listeners;              // 0x0168 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
float                      HighestScorability;       // 0x0178 (0x0004)
[0x0000004000000000]
float                      ClosestBallToGoal;        // 0x017C (0x0004)
[0x0000004000000000]
struct FCarData             HighestAerialCar;        // 0x0180 (0x0058)
[0x0000004000400000] (CPF_NeedCtorLink)
struct FGoalData             ClosestGoalToBall;       // 0x01D8 (0x0048)
[0x0000004000080000] (CPF_Component)
struct FScriptDelegate      __EventCarHitBall__Delegate; // 0x0220
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameObserver_TA");
}

return uClassPointer;
};

void __GameObserver_TA__InitCars_0x1(class ACar_TA* C);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void eventDrawGoalData(struct FGoalData Data);
void eventDrawCarData(struct FCarData Data);
void eventDrawBallData(struct FBallData Data);
float GetAerialSkill(float Altitude);
struct FVector GetLocationAtTime(class ARBActor_TA* InActor, float inTime);
struct FVector ClampLocationToGoal(class UGoal_TA* InGoal, struct FVector InLocation);
void GetCarGoalData(class AVehicle_TA* ForCar, struct FGoalData& ForGoal, struct FCarGoalData& Data);
struct FGoalData GetOffensiveGoal(class AVehicle_TA* ForCar);

```

```

struct FGoalData GetDefensiveGoal(class AVehicle_TA* ForCar);
void UpdateCarData(float DeltaTime, struct FCarData& Data);
struct FCarData GetCarDataForPRI(class APRI_TA* PRI);
struct FCarData GetCarData(class AVehicle_TA* ForCar);
void UpdateCarsData(float DeltaTime);
void UpdateGoalData(float DeltaTime, struct FGoalData& Data);
void UpdateGoalsData(float DeltaTime);
void UpdateBallData();
void eventTick(float DeltaTime);
class AGameEvent_Soccar_TA* GetSoccarEvent();
class ABall_TA* GetBall();
void HandleCarHitBall(class ACar_TA* InCar, class ABall_TA* InBall, struct FVector HitLocation,
struct FVector HitNormal);
void HandleCarRemoved(class AGameEvent_TA* G, class ACar_TA* InCar);
void HandleCarAdded(class AGameEvent_TA* G, class ACar_TA* InCar);
void InitCars();
void HandleBallRemoved(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* InBall);
void HandleBallAdded(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* InBall);
void InitBall();
void InitFieldProperties();
void eventAttached();
void EventCarHitBall(class ACar_TA* InCar, class ABall_TA* InBall);
};

// Class TAGame.GameShare_TA
// 0x00D8 (0x0060 - 0x0138)
class UGameShare_TA : public UObject
{
public:
class UMaxActorsGroup_TA*           CarEngineAudioGroup;          // 0x0060
(0x0008) [0x00000000408000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<class UMaterialInstance*>     BallPositionMaterialInstances; // 0x0068
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class AVehiclePickup_Boost_TA*> ActiveBoostPills;        // 0x0078
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long                      bHideWorldUI : 1;           // 0x0088 (0x004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
TArray<class UNameplateComponent_TA*> Nameplates;              // 0x0090
(0x0010) [0x0000004004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
TArray<class ACarPreviewActor_TA*>   CarPreviewActors;          // 0x00A0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UAkSoundSource*>        SoundSourcePool;          // 0x00B0
(0x0010) [0x000000000448200A] (CPF_Const | CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
struct FScriptDelegate             _EventHideWorldUIChanged__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             _EventNameplateAdded__Delegate;  // 0x00D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             _EventNameplateRemoved__Delegate; // 0x00F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             _EventCarPreviewActorAdded__Delegate; // 0x0108
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

struct FScriptDelegate           _EventCarPreviewActorRemoved__Delegate;    //
0x0120 (0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameShare_TA");
}

return uClassPointer;
};

void NotifyCarPreviewActorAdded(struct FScriptDelegate Callback);
void RemoveCarPreviewActor(class ACarPreviewActor_TA* PreviewCar);
void AddCarPreviewActor(class ACarPreviewActor_TA* PreviewCar);
void RemoveNameplate(class UNameplateComponent_TA* Nameplate);
void AddNameplate(class UNameplateComponent_TA* Nameplate);
void SetWorldUIHidden(unsigned long bHidden);
void NotifyHideWorldUIChanged(struct FScriptDelegate Callback);
void EventCarPreviewActorRemoved(class UGameShare_TA* GameShare, class
ACarPreviewActor_TA* PreviewCar);
void EventCarPreviewActorAdded(class UGameShare_TA* GameShare, class
ACarPreviewActor_TA* PreviewCar);
void EventNameplateRemoved(class UGameShare_TA* GameShare, class
UNameplateComponent_TA* NameplateComponent);
void EventNameplateAdded(class UGameShare_TA* GameShare, class
UNameplateComponent_TA* NameplateComponent);
void EventHideWorldUIChanged(class UGameShare_TA* GameShare);
class UAkSoundSource* GetPooledSoundSource();
};

// Class TAGame.GameViewportClient_TA
// 0x00F8 (0x02C8 - 0x03C0)
class UGameViewportClient_TA : public UGameViewportClient_X
{
public:
unsigned long          bSplitScreenDisabled : 1;           // 0x02C8 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bUpdateSplitscreenSizes : 1;        // 0x02C8 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long          bDrawWorldFullScreen : 1;           // 0x02C8 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
unsigned long          bDrawBuildInfo : 1;                 // 0x02C8 (0x0004)
[0x0000000000002000] [0x00000008] (CPF_Transient)
TArray<float>          LastSplitJoinTryTimes;           // 0x02D0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
float                  UIScaleModifier;                   // 0x02E0 (0x0004)
[0x0000000000000000]
float                  NameplateScaleModifier;           // 0x02E4 (0x0004)
[0x0000000000000000]

```

```

class AGameEvent_TA*           GameEvent;          // 0x02E8 (0x0008)
[0x0000800000000000]
class UOnlineGame_X*          OnlineGame;        // 0x02F0 (0x0008)
[0x0000800000000000]
class FString                 SplitscreenPlayerLeftTournamentTitle; // 0x02F8
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                 SplitscreenPlayerLeftTournamentBody; // 0x0308
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate         __EventPostInit__Delegate; // 0x0318 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventSplitScreenTypeChanged__Delegate; // 0x0330
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventDrawWorldFullScreenChanged__Delegate; // 0x0348 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventUIScaleModifierUpdated__Delegate; // 0x0360
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventSafeZoneRatioUpdated__Delegate; // 0x0378
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventUpdatedNumLocalPlayers__Delegate; // 0x0390 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventNameplateScaleModifierUpdated__Delegate; // 0x03A8 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameViewportClient_TA");
}

return uClassPointer;
};

void __GameViewportClient_TA_Init_0x4();
void __GameViewportClient_TA_Init_0x3(class AGameEvent_TA* _);
void __GameViewportClient_TA_Init_0x2(class ULocalPlayer* _);
void __GameViewportClient_TA_Init_0x1(class ULocalPlayer* _);
int32_t __GameViewportClient_TA__AddSplitScreenPlayer_0x1(class ULocalPlayer* LP);
void eventSetProgressMessage(uint8_t MessageType, class FString Message, class FString Title, unsigned long bIgnoreFutureNetworkMessages);
void CheckUnsubscribeTournament();
void NotifyPlayerRemoved(int32_t PlayerIndex, class ULocalPlayer* RemovedPlayer);
void SetNameplateScaleModifier(float Value);
void SetUIScaleModifier(float Value);
void SetSafeZone(float Value);
void HandlePrimaryPlayerChange(class ULocalPlayer* OldPrimary, class ULocalPlayer* NewPrimary);
static int32_t GetLocalPlayerCount();
void NotifyOnNumLocalPlayersUpdated(struct FScriptDelegate Callback);
void UpdateNumLocalPlayers();
float GetAspectRatio(class ULocalPlayer* Player);

```

```

class ULocalPlayer* eventCreatePlayer(int32_t ControllerId, unsigned long bSpawnActor, class FString& OutError);
void AddSplitScreenPlayer();
void HandleUserSignInComplete_CreateSplitScreenPlayer(uint8_t LocalUserNum);
void InitiateAsyncLoginForSplitscreen(int32_t ControllerId);
bool HandleSplitScreenJoinAsync(int32_t ControllerId);
bool HandleKeyPress(int32_t ControllerId, struct FName Key, uint8_t EventType, float AmountDepressed, unsigned long bGamepad);
void SplitscreenJoinResponse(struct FUniqueNetId PlayerID, unsigned long bAllow, class FString Error);
void RequestSplitscreenJoin(int32_t ControllerId);
bool AllowTournamentSplitscreen(int32_t ControllerId);
bool GameEventAllowSplitScreenPlayer();
void CreateSplitScreenPlayer(int32_t ControllerId);
void HandleSplitscreenJoinPrivilegeCheck(class UPrivilegeCheck_X* PrivilegeCheck);
void TryToCreateSplitScreenPlayer(int32_t ControllerId);
void HandleUserSwitchComplete(uint8_t LocalUserNum);
bool IsPlayingOnline();
bool GameWantsFullScreen();
void eventLayoutPlayers();
void SetSplitscreenSizes();
class APlayerControllerBase_TA* GetPrimaryPlayerController();
void UpdateActiveSplitscreenType();
void DrawTransition(class UCanvas* Canvas);
void NotifyConnectionError(uint8_t MessageType, class FString Message, class FString Title);
void HandleExternalUIChanged(unsigned long blsOpening);
bool eventInit(class FString& OutError);
void EventNameplateScaleModifierUpdated(class UGameViewportClient_TA* GVC);
void EventUpdatedNumLocalPlayers(class UGameViewportClient_TA* GVC, int32_t LocalPlayerCount);
void EventSafeZoneRatioUpdated(class UGameViewportClient_TA* GVC);
void EventUIScaleModifierUpdated(class UGameViewportClient_TA* GVC);
void EventDrawWorldFullScreenChanged(class UGameViewportClient_TA* GVC);
void EventSplitScreenTypeChanged(class UGameViewportClient_TA* GVC);
void EventPostInit(class UGameViewportClient_TA* GVC);
};

// Class TAGame.GFxData_DevUtil_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_DevUtil_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_DevUtil_TA");
}

return uClassPointer;
}

```

```

};

bool Assert(unsigned long expr, class FString Message);
void Log_Native(class FString Category, class FString Message);
void Log(class FString Category, class FString Message);
};

// Class TAGame.GFxData_Personas_TA
// 0x0058 (0x0098 - 0x00F0)
class UGFxData_Personas_TA : public UGFxDataSingleton_X
{
public:
    struct FMap_Mirror           PersonaMap;          // 0x0098 (0x0050)
    [0x0000000000001000] (CPF_Native)
    class UPersonas_TA*          PersonasData;        // 0x00E8 (0x0008)
    [0x0000804000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_Personas_TA");
        }

        return uClassPointer;
    };
};

int32_t GetOrCreatePersonaDataFromID(struct FUniqueNetId PlayerID);
class UGFxData_Persona_TA* GetOrCreatePersonaData(class UPersona_TA* Persona);
void HandlePartyMemberAdded(class UOnlineGameParty_X* PartyObject, struct FUniqueNetId InMemberId);
void HandleLocalPlayerAdded(class ULocalPlayer_TA* InLocalPlayer);
static class UGFxData_Persona_TA* GetOrCreateFromID(class UGFxShell_X* InShell, struct FUniqueNetId PlayerID);
static class UGFxData_Persona_TA* GetOrCreate(class UGFxShell_X* InShell, class UPersona_TA* Persona);
void eventOnRemoved();
void eventOnShellSet();
int32_t GetFromMap(struct FUniqueNetId PlayerID);
void AddToMap(struct FUniqueNetId PlayerID, int32_t RowIdx);
};

// Class TAGame.GFxData_ShopCatalogue_TA
// 0x00EC (0x0094 - 0x0180)
class UGFxData_ShopCatalogue_TA : public UGFxDataRow_X
{
public:
    class UShopsManager_TA*      ShopManager;         // 0x0098 (0x0008)
    [0x0000800000000001] (CPF_Edit)
    int32_t                      ShopID;              // 0x00A0 (0x0004)
    [0x0000000040003000] (CPF_Native | CPF_Transient | CPF_EditInlineNotify)
};

```

```
class FString           ShopName;          // 0x00A8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString           Type;             // 0x00B8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString           Region;           // 0x00C8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
uint64_t                StartDate;        // 0x00D8 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
uint64_t                EndDate;          // 0x00E0 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
class UTexture*         ShopImage;        // 0x00E8 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
class FString           ShopImageURL;     // 0x00F0 (0x0010)
[0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long            bLoaded : 1;       // 0x0100 (0x0004)
[0x0008000040002000] [0x00000001] (CPF_Transient | CPF>EditInlineNotify)
unsigned long            bNeedsUpdate : 1;   // 0x0100 (0x0004)
[0x001000040002000] [0x00000002] (CPF_Transient | CPF>EditInlineNotify)
unsigned long            bRelatedCatalogueBatchRequestInProgress : 1; // 0x0100
(0x0004) [0x00100000002000] [0x00000004] (CPF_Transient)
unsigned long            bFailedToLoad : 1;    // 0x0100 (0x0004)
[0x0000000040002000] [0x00000008] (CPF_Transient | CPF>EditInlineNotify)
unsigned long            bShopItemExpired : 1;  // 0x0100 (0x0004)
[0x0000000040002000] [0x00000010] (CPF_Transient | CPF>EditInlineNotify)
unsigned long            bAutoReload : 1;      // 0x0100 (0x0004)
[0x0008000040002000] [0x00000020] (CPF_Transient | CPF>EditInlineNotify)
unsigned long            bNewCatalogue : 1;    // 0x0100 (0x0004)
[0x0000000040002000] [0x00000040] (CPF_Transient | CPF>EditInlineNotify)
TArray<struct FShopItem> CachedItems;      // 0x0108 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
float                  CacheDuration;     // 0x0118 (0x0004)
[0x0000000000004000] (CPF_Config)
float                  LastCacheTime;     // 0x011C (0x0004)
[0x0000000000002000] (CPF_Transient)
float                  LastOwnedCacheTime; // 0x0120 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                  LastOwnershipDesyncTime; // 0x0124 (0x0004)
[0x00100000002000] (CPF_Transient)
class UAsyncTask*       ShopTask;          // 0x0128 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UOnlineProductStoreSet_TA* CatalogueSet; // 0x0130
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
CPF>EditInline)
class UShopItemExpirationHelper_TA* ExpirationHelper; // 0x0138
(0x0008) [0x0000000000002000] (CPF_Transient)
class UShopAutoLoadHelper_TA* AutoLoadHelper; // 0x0140
(0x0008) [0x0000000000002000] (CPF_Transient)
class UShopsConfig_TA* ShopsConfig; // 0x0148 (0x0008)
[0x0000800000000000]
struct FScriptDelegate   __bLoaded__ChangeNotify; // 0x0150
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate   __bAutoReload__ChangeNotify; // 0x0168
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ShopCatalogue_TA");
}

return uClassPointer;
};

void __GFxData_ShopCatalogue_TA__OnShellSet_0x1(class UShopItemExpirationHelper_TA* _);
int32_t __GFxData_ShopCatalogue_TA__ContainsOnlineProducts_0x2(class UOnlineProduct_TA* P);
int32_t __GFxData_ShopCatalogue_TA__ContainsOnlineProducts_0x1(struct FOnlineProductData P);
struct FProductHashID __GFxData_ShopCatalogue_TA__GetOnlineProductsByCostID_0x2(struct FShopDeliverable D);
bool __GFxData_ShopCatalogue_TA__SetCataloguelItems_0x1(struct FShopDeliverable P);
bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x3(struct FShopDeliverable D);
bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x2(struct FShopDeliverable P);
bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x1(class UOnlineProduct_TA* P);
void __bAutoReload_ChangeNotifyFunc();
void __bLoaded_ChangeNotifyFunc();
bool IsActive(uint64_t CurrentTime);
void DebugSetCataloguelItems(TArray<struct FShopItem> InItems);
void DebugOverride(int32_t Index, int32_t ProductID, int32_t PaintID, int32_t CertifiedId, int32_t SpecialEditionID, int32_t Cost);
void OnCatalogueLoaded();
void HandleShopItemExpired();
void HandleCatalogueLoadFailed(class UError* Error);
void ShopItemTextureLoaded(int32_t ShopItemID, class UTexture* LoadedTexture);
void SyncItemImages(class UWebImageCache_X* WebImageCache, int32_t I);
void HandleCatalogueLoaded(TArray<struct FShopItem> InItems, float RequestStartTime);
void SetIsNewCatalogue(unsigned long bInNewCatalogue);
void SetOwnedDeliverableProducts();
bool IsDeliverableSubset(TArray<struct FShopDeliverable>& Superset, TArray<struct FShopDeliverable>& Subset);
void MarkOwnershipDirty();
void ClearLastOwnedCacheTime();
void SetCataloguelItems(TArray<struct FShopItem> InItems, float RequestStartTime);
bool GetShopItemByCostID(int32_t CostID, struct FShopItem& OutShopItem);
TArray<class UOnlineProduct_TA*> GetOnlineProductsByCostID(int32_t CostID);
bool ContainsOnlineProducts(TArray<struct FOnlineProductData>& Products);
void LoadCatalogueInternal(unsigned long bForce);
bool IsCatalogueRequestInProgress();
bool ShouldRequestCatalogueLoad();
void NotifyCatalogueBatchLoadSuccess(TArray<struct FShopItem> InItems, float RequestStartTime);
```

```

void NotifyCatalogueBatchLoadFailure(class UError* Error);
void NotifyBatchLoadStarted();
void SetAutoLoadCatalogue(unsigned long bInAutoReload);
TArray<struct FShopItem> GetShopItems();
void ForceLoadCatalogue();
void LoadCatalogue();
void Init(struct FShopData InData);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_Shops_TA
// 0x00D0 (0x0098 - 0x0168)
class UGFxData_Shops_TA : public UGFxDataSingleton_X
{
public:
    class UShopsManager_TA*           ShopManager;           // 0x0098 (0x0008)
    [0x0000800000000001] (CPF_Edit)
    class UOnlineImageDownloaderWeb*   ImageDownloader;       // 0x00A0
    (0x0008) [0x0000800000000001] (CPF_Edit)
    class UShopsConfig_TA*            Config;                // 0x00A8 (0x0008)
    [0x000080000002000] (CPF_Transient)
    TArray<class UGFxData_ShopCatalogue_TA*>   Shops;          // 0x00B0
    (0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    struct FMap_Mirror               ShopsMap;              // 0x00C0 (0x0050)
    [0x000100000003000] (CPF_Native | CPF_Transient)
    int32_t                          DebugShopID;          // 0x0110 (0x0004)
    [0x000000040000000] (CPF>EditInlineNotify)
    unsigned long                     bHideMainMenuButton : 1; // 0x0114 (0x0004)
    [0x000000040000000] [0x00000001] (CPF>EditInlineNotify)
    unsigned long                     bNewCatalogues : 1;  // 0x0114 (0x0004)
    [0x000000040000000] [0x00000002] (CPF>EditInlineNotify)
    class UOnlineProductStoreSet_TA*  CrateShopSet;         // 0x0118
    (0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
    float                            CacheDuration;        // 0x0120 (0x0004)
    [0x000000000004000] (CPF_Config)
    float                            LastCacheTime;        // 0x0124 (0x0004)
    [0x000000000002000] (CPF_Transient)
    class UAsyncTask*                ShopTask;              // 0x0128 (0x0008)
    [0x000000000002000] (CPF_Transient)
    class UShopMetrics_TA*           ShopMetrics;           // 0x0130 (0x0008)
    [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
    class UShopTabs_TA*              ShopTabs;              // 0x0138 (0x0008)
    [0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
    class UShopNotificationsManager_TA*  ShopNotificationsManager; // 0x0140 (0x0008) [0x000004000000000]
    class UShopCatalogueCacheSave_TA* CatalogueCache;        // 0x0148
    (0x0008) [0x000008000002000] (CPF_Transient)
    struct FScriptDelegate           _EventReceivedShops__Delegate; // 0x0150
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Shops_TA");
}

return uClassPointer;
};

void __GFxData_Shops_TA__OnShellSet_0x1(class UShopCatalogueCacheSave_TA* InCache);
class UGFxData_ShopCatalogue_TA* __GFxData_Shops_TA__BuildShopCatalogues_0x1(struct FShopData Data);
class FString __GFxData_Shops_TA__HandleGetAllShops_0x2(struct FShopData S);
bool __GFxData_Shops_TA__HandleGetAllShops_0x1(struct FShopData S);
bool __GFxData_Shops_TA__AnyCatalogueNeedsUpdate_0x1(class UGFxData_ShopCatalogue_TA* S);
int32_t __GFxData_Shops_TA__LoadCataloguesByID_0x1(int32_t Id);
bool __GFxData_Shops_TA__RefreshShopsViewed_0x1(class UGFxData_ShopCatalogue_TA* S);
bool __GFxData_Shops_TA__SyncFeaturedProducts_0x1(class UGFxData_ShopCatalogue_TA* S);
void __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x3(class UGFxModal_X* _);
void __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x2(class UGFxModal_X* _);
bool __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x1(class UOnlineProduct_TA* P);
TArray<struct FShopTab> GetAvailableShopTabs();
static void CreateTemporaryOnlineProducts(class UOnlineProductStoreSet_TA* ProductSet,
TArray<struct FShopItem>& Items);
void OnGetCrateShopFailed();
void HandleGetCrateShopFailed(class UError* Error);
void OnReceivedCrateShop(int32_t CrateID, TArray<struct FShopItem> CrateItems, int32_t ShopID);
void GetCrateShop(int32_t CrateID);
void OnPurchaseFail();
void HandlePurchaseFail(class UError* Error);
void OnPurchaseSuccess();
void HandlePurchaseSuccess(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count,
TArray<struct FOnlineProductData> PurchasedProducts, TArray<struct FCurrency> PurchasedCurrencies);
void OnPurchase(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count, int32_t ExpectedPriceID, int32_t ExpectedPriceAmount, int32_t ExpectedProductCount);
void OnCheckForPurchaseWarningSuccess();
bool OnCheckForPurchaseWarning(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count);
void CheckForPurchaseWarning(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count);
bool CheckPlayerMeetsProductRequirements(class UOnlineProduct_TA* OnlineProduct,
TArray<class UOnlineProduct_TA*>& DeliverableOnlineProducts, class FString& ErrorString);
void HandleShopItemTextureLoaded(int32_t ShopID, int32_t ShopItemIndex, class UTexture* LoadedTexture);
void Purchase(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count, int32_t ExpectedPriceID, int32_t ExpectedPriceAmount, int32_t ExpectedProductCount);
void PreviewShopItem(int32_t ShopItemID);
void ExitShopPage();

```

```

void OnMTXShopChanged(int32_t NewCatalogID);
void OnShopCatalogueChanged(int32_t NewShopID);
void OnEnterShopMenu(int32_t BrowseSourceID);
void SyncFeaturedProducts(TArray<struct FOnlineProductData>& Products);
void RefreshOwnedProducts();
void OnShopCataloguesOwnershipDesync(unsigned long bTrackedCatalogueMarkedDirty);
void RefreshShopsViewed();
void SetShopViewed(int32_t InShopID);
int32_t GetShopCatalogueIndex(int32_t InShopID);
bool CheckShopExists(int32_t InShopID);
void OnGetAllShopsFailed();
void OnGetAllShopsSuccess();
void HandleGetShopCataloguesFailed(TArray<int32_t> ShopIDs, class UError* Error);
void HandleGetShopCataloguesSuccess(TArray<int32_t> ShopIDs, float RequestStartTime,
TArray<struct FShopCatalogue>& RequestCatalogues);
void LoadCataloguesByIndex(TArray<int32_t> ShopIndices);
void LoadCataloguesByID(TArray<int32_t> ShopIDs);
void LoadAllShopCatalogues(int32_t ForcedCatalogueID);
bool AnyCatalogueNeedsUpdate();
void HandleGetAllShopsFailed(class UError* Error);
void HandleImageDownloaded(struct FOnlineImageDownload DownloadResult);
void HandleGetAllShops(TArray<struct FShopData> InShops);
TArray<class UGFxData_ShopCatalogue_TA*> BuildShopCatalogues(TArray<struct FShopData>&
InShops);
void GetAllShopsInternal(unsigned long bForce);
void GetAllShopsForced();
void RefreshAllShops();
void HandlePsyNetLogin();
void HandleConfigChanged();
void eventOnRemoved();
void eventOnShellSet();
struct FShopsMapEntry GetCatalogueFromShopID(int32_t ShopID);
void BuildShopsMap(TArray<class UGFxData_ShopCatalogue_TA*>& InShops);
void EventReceivedShops(class UGFxData_Shops_TA* GFxShops);
};

// Class TAGame.GFxEngine_TA
// 0x0028 (0x01A0 - 0x01C8)
class UGFxEngine_TA : public UGFxEngine_X
{
public:
TArray<struct FSoundStateItem>           UISoundStates;          // 0x01A0
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long                bViewingOnlineScene : 1;        // 0x01B0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
class ULoadingScreen_TA*                 LoadingScreen;        // 0x01B8 (0x0008)
[0x0000000000000000]
class UOnlineSubsystem*                  OnlineSubsystem;      // 0x01C0 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxEngine_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
int32_t UISoundStatePrioritySort(struct FSoundStateItem A, struct FSoundStateItem B);
void UpdateUISoundState();
void OnShellDestroyed(class UGFxShell_X* Shell);
void eventOnGameSessionEnded();
void OnExternalUIChanged(unsigned long bIsOpening);
uint8_t GetSoundStatePriority(struct FName StateName);
void ClearUISoundState(class UGFxShell_X* Shell);
void PopUISoundState(struct FName StateName, class UGFxShell_X* Shell);
void PushUniqueUISoundState(struct FName StateName, class UGFxShell_X* Shell);
void PushUISoundState(struct FName StateName, class UGFxShell_X* Shell);
void InitOnlineSub();
void Init();
};

// Class TAGame.GlobalLookupTexture_TA
// 0x007C (0x01B8 - 0x0234)
class UGlobalLookupTexture_TA : public UScriptedTexture
{
public:
struct FltVector           EnvironmentAPosition;          // 0x01B8 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FltVector           EnvironmentBPosition;         // 0x01C0 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FltVector           SunlightPosition;            // 0x01C8 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FltVector           WorldLightPosition;          // 0x01D0 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
int32_t                   TeamsPositionX;              // 0x01D8 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
int32_t                   PlayersPositionX;            // 0x01DC (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
TArray<struct FLinearColor> TopTeamColors;             // 0x01E0 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FLinearColor> BottomTeamColors;          // 0x01F0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UObject*>    Players;                    // 0x0200 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FLinearColor> TopPlayerColors;           // 0x0210 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FLinearColor> BottomPlayerColors;         // 0x0220
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long              bDebug : 1;                  // 0x0230 (0x0004)
[0x0000000000000000] [0x00000001]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GlobalLookupTexture_TA");
}

return uClassPointer;
};

void OnRender(class UCanvas* Canvas);
static struct FLinearColor GetFullBrightColor(struct FLinearColor InColor, float ValueScale);
void Update();
void SetTeamColors(int32_t Team, struct FLinearColor Primary, struct FLinearColor Secondary);
void RemovePlayer(class UObject* Player);
float AddPlayer(class UObject* Player, struct FLinearColor Primary, struct FLinearColor Secondary);
};

// Class TAGame.Goal_TA
// 0x011C (0x00A4 - 0x01C0)
class UGoal_TA : public UActorComponent_X
{
public:
class AActor* GoalOrientation; // 0x00A8 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class AActor* ReplayOrientation; // 0x00B0 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
TArray<class AActor*> OverrideGoalIndicatorOrientations; // 0x00B8
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FVector SaveCheckDirectionOffset; // 0x00C8 (0x000C)
[0x0000000000000001] (CPF_Edit)
float EpicSaveMinSpeed; // 0x00D4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float EpicSaveArriveTime; // 0x00D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
uint8_t TeamNum; // 0x00DC (0x0001)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X* ScoreFX; // 0x00E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class FString GoalIndicatorArchetype; // 0x00E8 (0x0010)
[0x00008000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
unsigned long bNoGoalIndicator : 1; // 0x00F8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bOnlyGoalsFromDirection : 1; // 0x00F8 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bAllowPostMatchLobbyReset : 1; // 0x00F8
(0x0004) [0x0000004000000001] [0x00000004] (CPF_Edit)
unsigned long bEnabled : 1; // 0x00F8 (0x0004)
[0x0000080000000001] [0x00000008] (CPF_Edit)
unsigned long bShowFocusExtent : 1; // 0x00F8 (0x0004)

```

```

[0x0000000000000001] [0x00000010] (CPF_Edit)
class AActor* GoalDirection; // 0x0100 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
int32_t PointsToAward; // 0x0108 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector AutoCamFocusExtent; // 0x010C (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector GoalFocusLocationOffset; // 0x0118 (0x000C)
[0x0000000000000001] (CPF_Edit)
float MaxGoalScorerAttachRadius; // 0x0124 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector GoalScoredDotDirection; // 0x0128 (0x000C)
[0x0000000000000001] (CPF_Edit)
float MinAttachGoalToScorerDot; // 0x0134 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector Location; // 0x0138 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector Direction; // 0x0144 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector Right; // 0x0150 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector Up; // 0x015C (0x000C)
[0x000008000002000] (CPF_Transient)
struct FRotator Rotation; // 0x0168 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector LocalExtent; // 0x0174 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector WorldCenter; // 0x0180 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector WorldExtent; // 0x018C (0x000C)
[0x000008000002000] (CPF_Transient)
struct FVector WorldFrontCenter; // 0x0198 (0x000C)
[0x000008000002000] (CPF_Transient)
struct FBox WorldBox; // 0x01A4 (0x001C)
[0x000008000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Goal_TA");
}

return uClassPointer;
};

void SetEnabled(unsigned long bValue);
struct FVector GetSaveCheckDirection();
struct FVector GetGoalFocusExtentCenter();
class AActor* GetReplayOrientation();
class AActor* GetGoalOverrideOrientationFromBallLocation(class ABall_TA* Ball);

```

```
void Init();
void eventBeginPlay();
};

// Class TAGame.GoalVolume_TA
// 0x000C (0x02A4 - 0x02B0)
class AGoalVolume_TA : public AVolume
{
public:
class UGoal_TA* Goal; // 0x02A8 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GoalVolume_TA");
}

return uClassPointer;
};

};

// Class TAGame.GroundStaticMeshActor_TA
// 0x0004 (0x0288 - 0x028C)
class AGroundStaticMeshActor_TA : public AStaticMeshActor
{
public:
float HideLocationOffset; // 0x0288 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GroundStaticMeshActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.HandlingPreset_TA
// 0x0080 (0x0060 - 0x00E0)
class UHandlingPreset_TA : public UObject
{
```

```
public:
struct FVehicleAxeData           FrontAxe;          // 0x0060 (0x0014)
[0x0000000000000001] (CPF_Edit)
struct FVehicleAxeData           BackAxe;          // 0x0074 (0x0014)
[0x0000000000000001] (CPF_Edit)
class UPhysicsAsset*             PhysAsset;         // 0x0088 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FSimpleSpringSettings     ChassisSpringDefaults; // 0x0090
(0x0050) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.HandlingPreset_TA");
}

return uClassPointer;
};

};

// Class TAGame.HideLevelAssetsBase_TA
// 0x0020 (0x0268 - 0x0288)
class AHideLevelAssetsBase_TA : public AActor
{
public:
TArray<class AActor*>          ActorsInLevel;      // 0x0268 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class UClass*>          ActorClasses;       // 0x0278 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.HideLevelAssetsBase_TA");
}

return uClassPointer;
};

void FindLevelActors();
};

// Class TAGame.GreenScreen_TA
// 0x0014 (0x0288 - 0x029C)
class AGreenScreen_TA : public AHideLevelAssetsBase_TA
```

```
{  
public:  
TArray<class UPrimitiveComponent*> HiddenPrimitives; // 0x0288  
(0x0010) [0x000000004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink |  
CPF_EditInline)  
unsigned long bOn : 1; // 0x0298 (0x0004)  
[0x0000000000002000] [0x00000001] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GreenScreen_TA");  
}  
  
return uClassPointer;  
};  
  
void ChangeBackgroundColor(float R, float G, float B);  
void End(unsigned long bUseSystemSettings);  
void Start(float R, float G, float B, unsigned long bUseSystemSetting);  
};  
  
// Class TAGame.ToggleActorVisibility_TA  
// 0x0000 (0x0288 - 0x0288)  
class AToggleActorVisibility_TA : public AHideLevelAssetsBase_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ToggleActorVisibility_TA");  
}  
  
return uClassPointer;  
};  
  
void ToggleActorVisibility(int32_t Index);  
};  
  
// Class TAGame.ICanEquip_TA  
// 0x0000 (0x0060 - 0x0060)  
class UICanEquip_TA : public UInterface  
{  
public:
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ICanEquip_TA");
}

return uClassPointer;
};

bool CanEquip(TArray<int32_t>& EquippedProducts);
};

// Class TAGame.IdenticalProductCache_TA
// 0x0050 (0x0070 - 0x00C0)
class UIdenticalProductCache_TA : public UComponent
{
public:
struct FMap_Mirror           Products;           // 0x0070 (0x0050)
[0x0000000000003002] (CPF_Const | CPF_Native | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdenticalProductCache_TA");
}

return uClassPointer;
};

TArray<class UOnlineProduct_TA*> FindProductsWithIdenticalAttributes(class
UOnlineProduct_TA* OnlineProduct, TArray<class UClass*> AttributesToIgnore);
class UOnlineProduct_TA* FindProduct(struct FProductInstanceID InstanceID);
class UOnlineProduct_TA* FindProductFromHash(struct FProductHashID HashID);
TArray<class UOnlineProduct_TA*> GetProductsFromHash(struct FProductHashID HashID);
TArray<class UOnlineProduct_TA*> MapHashIDs(TArray<struct FProductHashID>& HashIDs);
TArray<class UOnlineProduct_TA*> GetUniqueProducts();
TArray<class UOnlineProduct_TA*> GetAllProducts();
TArray<struct FProductInstanceID> GetAllInstanceIDs();
TArray<class UOnlineProduct_TA*> GetIdentical(class UOnlineProduct_TA* Product);
int32_t GetQuantity(class UOnlineProduct_TA* Product, unsigned long bHoldXEInstance);
int32_t GetNumProducts();
bool HasAnyProducts();
void Clear();
void RemoveInstanceId(struct FProductInstanceID InstanceID);
void Remove(class UOnlineProduct_TA* OnlineProduct);
void Add(class UOnlineProduct_TA* OnlineProduct);
```

```

void Append(class UIdenticalProductCache_TA* ProductCache);
void Set(TArray<class UOnlineProduct_TA*>& OnlineProducts);
};

// Class TAGame.ImpactEffectsComponent_TA
// 0x0154 (0x00A4 - 0x01F8)
class UImpactEffectsComponent_TA : public UActorComponent_X
{
public:
int32_t SoundPriority; // 0x00A8 (0x0004)
[0x0000000000000001] (CPF_Edit) class UEffectsMap_X* ImpactEffectsMap; // 0x00B0 (0x0008)
[0x0000000000000001] (CPF_Edit) class UAkSoundCue* AkImpactSound; // 0x00B8 (0x0008)
[0x0000000000000001] (CPF_Edit) class UAkSoundCue* AkSlideSound; // 0x00C0 (0x0008)
[0x0000000000000001] (CPF_Edit) float AkSlideSoundDelay; // 0x00C8 (0x0004)
[0x0000000000000001] (CPF_Edit) float AkSlideMomentumMin; // 0x00CC (0x0004)
[0x0000000000000001] (CPF_Edit) struct FName AkImpactTypeKey; // 0x00D0 (0x0008)
[0x0000000000000001] (CPF_Edit) struct FName AkImpactIntensityKey; // 0x00D8 (0x0008)
[0x0000000000000001] (CPF_Edit) float MinImpactMomentum; // 0x00E0 (0x0004)
[0x0000000000000001] (CPF_Edit) float MinImpactDelay; // 0x00E4 (0x0004)
[0x0000000000000001] (CPF_Edit) class UCameraShake* ImpactCameraShake; // 0x00E8
(0x0008) [0x0000000000000001] (CPF_Edit) class UForceFeedbackWaveform* ImpactForceFeedback; // 0x00F0
(0x0008) [0x0000000000000001] (CPF_Edit) struct FInterpCurveFloat ShakeScaleCurve; // 0x00F8 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink) float LastWorldImpactTime; // 0x0110 (0x0004)
[0x0000000000002000] (CPF_Transient) float LastActorImpactTime; // 0x0114 (0x0004)
[0x0000000000002000] (CPF_Transient) float SlideTime; // 0x0118 (0x0004)
[0x0000000000002000] (CPF_Transient) class UParticleSystemComponent* SlideParticleComponent; // 0x0120
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline) class UAkPlaySoundComponent* SlideAudioComponent; // 0x0128
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline) class UShakeComponent_X* ShakeComponent; // 0x0130
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline) struct FCollisionEffectData LastCollisionEffectData; // 0x0138 (0x0058)
[0x0000000000002000] (CPF_Transient) struct FCollisionEffectData SlideCollisionEffectData; // 0x0190 (0x0058)
[0x0000000000002000] (CPF_Transient)

```

```

class UPhysicalMaterialProperty_TA*           SlidePhysMatProp;          // 0x01E8
(0x0008) [0x0000000000002000] (CPF_Transient)
class UArenaReflectionsManager_TA*           SlapbackManager;          // 0x01F0
(0x0008) [0x0008000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ImpactEffectsComponent_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
class UParticleSystemComponent* eventPlayPooledParticle(class UParticleSystem* Template,
struct FVector Location, struct FRotator Rotation, class AActor* AttachTo);
void HandleCollision(struct FAccumulatedRigidBodyCollision& Collision);
};

// Class TAGame.InputRecorder_TA
// 0x0044 (0x0268 - 0x02AC)
class AInputRecorder_TA : public AActor
{
public:
struct FPointer           VfTable_IITickNotify_TA;          // 0x0268 (0x0008)
[0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
float                     MaxRecordTime;                  // 0x0270 (0x0004)
[0x0000000000000001] (CPF>Edit)
unsigned long              bRecording : 1;                // 0x0274 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long              bPlaying : 1;                 // 0x0274 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
TArray<struct FReplicatedRBState>   CarStates;            // 0x0278 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FReplicatedRBState>   BallStates;           // 0x0288 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FVehicleInputs>       Inputs;               // 0x0298 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                      InputIndex;             // 0x02A8 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InputRecorder_TA");
}

```

```

}

return uClassPointer;
};

void eventFinished();
struct FReplicatedRBState eventGetBallState();
struct FReplicatedRBState eventGetCarState();
class ABall_TA* GetBall();
class ACar_TA* GetCar();
void StartPlayback();
void StopRecord();
void HandleDestroyed(class APawn_X* P);
void StartRecord();
};

// Class TAGame.InstanceIDUtils_TA
// 0x0000 (0x0060 - 0x0060)
class UInstanceIdUtils_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InstanceIDUtils_TA");
}

return uClassPointer;
};

static class FString ToHexString(struct FProductInstanceID InstanceID);
static uint64_t ToQWord(struct FProductInstanceID InstanceID);
static int32_tToInt(struct FProductInstanceID InstanceID);
static struct FProductInstanceID Subtract(struct FProductInstanceID InstanceID, uint64_t Value);
static int32_t SortDescending(struct FProductInstanceID A, struct FProductInstanceID B);
static int32_t SortAscending(struct FProductInstanceID A, struct FProductInstanceID B);
static struct FProductInstanceID Minimum(struct FProductInstanceID A, struct FProductInstanceID B);
static struct FProductInstanceID Maximum(struct FProductInstanceID A, struct FProductInstanceID B);
static bool IsValid(struct FProductInstanceID InstanceID);
static struct FProductInstanceID GetMaxInstanceID();
static TArray<struct FProductInstanceID> FromQWordArray(TArray<uint64_t>& InstanceIDs);
static struct FProductInstanceID FromHexString(class FString& InstanceID);
static struct FProductInstanceID FromQWord(uint64_t InstanceID);
static struct FProductInstanceID FromInt(int32_t InstanceID);
static int32_t Compare(struct FProductInstanceID A, struct FProductInstanceID B);
static struct FProductInstanceID Add(struct FProductInstanceID InstanceID, uint64_t Value);
};

```

```
// Class TAGame.InterpTrackAnimControlWithSensibleDefault_TA
// 0x0004 (0x0114 - 0x0118)
class UInterpTrackAnimControlWithSensibleDefault_TA : public UInterpTrackAnimControl
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.InterpTrackAnimControlWithSensibleDefault_TA");
}

return uClassPointer;
};

};

// Class TAGame.InterpTrackBoostGlow_TA
// 0x0004 (0x00E4 - 0x00E8)
class UInterpTrackBoostGlow_TA : public UInterpTrackFloatBase
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackBoostGlow_TA");
}

return uClassPointer;
};

};

// Class TAGame.InterpTrackBoostToggle_TA
// 0x0040 (0x00DC - 0x011C)
class UInterpTrackBoostToggle_TA : public UInterpTrackToggle
{
public:
class UProductAsset_Boost_TA*           OverrideBoost;          // 0x00E0
(0x0008) [0x0000000000000001] (CPF_Edit)
TArray<class UParticleSystemComponent*>   AttachedParticleSystems;    //
0x00E8 (0x0010) [0x000000004480008] (CPF_ExportObject | CPF_Component |
CPF_NeedCtorLink | CPF_EditInline)
```

```
TArray<struct FName> AttachedSocketNames; // 0x00F8
(0x0010) [0x0000000000040000] (CPF_NeedCtorLink)
class AActor* CurrentActor; // 0x0108 (0x0008)
[0x0000000000000000]
class AFXActor_Boost_TA* CurrentBoostFX; // 0x0110 (0x0008)
[0x0000000000000000]
unsigned long bIsBoostActive : 1; // 0x0118 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackBoostToggle_TA");
}

return uClassPointer;
};

};

// Class TAGame.InterpTrackCameraShake_TA
// 0x0020 (0x00DC - 0x00FC)
class UInterpTrackCameraShake_TA : public UInterpTrackToggle
{
public:
class UCameraShake* Shake; // 0x00E0 (0x0008)
[0x00000000400001] (CPF_Edit | CPF_EditInline)
float ShakeScale; // 0x00E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
uint8_t PlaySpace; // 0x00EC (0x0001)
[0x0000000000000001] (CPF_Edit)
struct FRotator PlaySpaceRotation; // 0x00F0 (0x000C)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackCameraShake_TA");
}

return uClassPointer;
};

void eventSetShakeActive(class UCameraModifier_CameraShake* InCameraShake, unsigned
long bActive);
};
```

```
// Class TAGame.InterpTrackCarScale_TA
// 0x0004 (0x00E4 - 0x00E8)
class UInterpTrackCarScale_TA : public UInterpTrackFloatBase
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackCarScale_TA");
}

return uClassPointer;
};

void eventSetTargetVehicleScale(class AVehicle_TA* TargetVehicle, float NewScale);
};

// Class TAGame.InterpTrackInstVehicle_TA
// 0x0008 (0x0060 - 0x0068)
class UInterpTrackInstVehicle_TA : public UInterpTrackInst
{
public:
class AVehicle_TA* TargetVehicle; // 0x0060 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackInstVehicle_TA");
}

return uClassPointer;
};

void eventInitTargetVehicleFX();
class AFXActor_Car_TA* GetTargetFX();
class AVehicle_TA* GetTargetVehicle();
};

// Class TAGame.InterpTrackInstVehicleToggle_TA
// 0x000C (0x006C - 0x0078)
class UInterpTrackInstVehicleToggle_TA : public UInterpTrackInstToggle
{
public:
```

```
class AVehicle_TA* TargetVehicle; // 0x0070 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackInstVehicleToggle_TA");
}

return uClassPointer;
};

void eventInitTargetVehicleFX();
class AFXActor_Car_TA* GetTargetFX();
class AVehicle_TA* GetTargetVehicle();
};

// Class TAGame.InterpTrackPostProcess_TA
// 0x02D4 (0x00DC - 0x03B0)
class UInterpTrackPostProcess_TA : public UInterpTrackToggle
{
public:
struct FPostProcessSettings SettingsToApply; // 0x00E0 (0x0168)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FPostProcessSettings PreviousCameraSettings; // 0x0248
(0x0168) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackPostProcess_TA");
}

return uClassPointer;
};

};

// Class TAGame.InterpTrackPostProcessIntensity_TA
// 0x0000 (0x00F0 - 0x00F0)
class UInterpTrackPostProcessIntensity_TA : public UInterpTrackFloatProp
{
public:

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackPostProcessIntensity_TA");
}

return uClassPointer;
};

};

// Class TAGame.InterpTrackSkelControlRotation_TA
// 0x000D (0x00E4 - 0x00F1)
class UInterpTrackSkelControlRotation_TA : public UInterpTrackFloatBase
{
public:
struct FName SkelControlName; // 0x00E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
uint8_t RotAxis; // 0x00F0 (0x0001)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackSkelControlRotation_TA");
}

return uClassPointer;
};

};

// Class TAGame.InterpTrackToggleDumpMovie_TA
// 0x0004 (0x00DC - 0x00E0)
class UInterpTrackToggleDumpMovie_TA : public UInterpTrackToggle
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackToggleDumpMovie_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.InterpTrackTriggerVehicleFX_TA
// 0x0014 (0x00DC - 0x00F0)
class UInterpTrackTriggerVehicleFX_TA : public UInterpTrackToggle
{
public:
struct FName AttachmentName; // 0x00E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UParticleSystemComponent* ParticleSystemInstance; // 0x00E8
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InterpTrackTriggerVehicleFX_TA");
}

return uClassPointer;
};

};

// Class TAGame.ItemDropGroup_TA
// 0x0060 (0x0060 - 0x00C0)
class UItemDropGroup_TA : public UObject
{
public:
TArray<struct FCurrency> Currency; // 0x0060 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FProductDrop> Products; // 0x0070 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t XP; // 0x0080 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString Message; // 0x0088 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UClass* GFxDataClass; // 0x0098 (0x0008)
[0x0000004000000002] (CPF_Const)
uint8_t DisplayOrder; // 0x00A0 (0x0001)
[0x0000000000000000]
struct FScriptDelegate _EventItemDropGroupHandled__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemDropGroup_TA");
}

return uClassPointer;
};

bool IsValid();
static void InsertGroup(class UItemDropGroup_TA* NewGroup, TArray<class
UItemDropGroup_TA*>& ItemDropGroups);
void EventItemDropGroupHandled(class UItemDropGroup_TA* ItemDropGroup);
};

// Class TAGame.ITickNotify_TA
// 0x0000 (0x0060 - 0x0060)
class UITickNotify_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ITickNotify_TA");
}

return uClassPointer;
};

};

// Class TAGame.LightCurveType_TA
// 0x0004 (0x0060 - 0x0064)
class ULightCurveType_TA : public UObject
{
public:
int32_t ParameterValue; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LightCurveType_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.LocalPlayer_TA
// 0x0080 (0x04E8 - 0x0568)
class ULocalPlayer_TA : public ULocalPlayer_X
{
public:
class USaveGameManager_TA*           SaveGameManager;          // 0x04E8
(0x0008) [0x0000008000000000]
class UProfile_TA*                 Profile;                // 0x04F0 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UObjectProvider*             ProfileObjectProvider; // 0x04F8 (0x0008)
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
uint8_t                           SplitScreenPosition;    // 0x0500 (0x0001)
[0x0000000000002000] (CPF_Transient)
uint64_t                          SuspendingEventToken;   // 0x0508 (0x0008)
[0x0000000000000000]
uint64_t                          SignoutStartedEventToken; // 0x0510 (0x0008)
[0x0000000000000000]
unsigned long                     bOpenedStartMenu : 1;   // 0x0518 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long                     bPendingBroadcastLoadout : 1; // 0x0518 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long                     bPendingBroadcastProfile : 1; // 0x0518 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
class UControllerLayoutStack*     ControllerLayouts;       // 0x0520 (0x0008)
[0x0001000000000000]
class UJoinGameConnectionChecker_TA* JoinGameConnectionChecker; // 0x0528 (0x0008) [0x0000000000002000] (CPF_Transient)
class UConnectionQualityConfig_TA* ConnectionQualityConfig; // 0x0530
(0x0008) [0x0000800000002000] (CPF_Transient)
struct FScriptDelegate           __EventProfileSet__Delegate; // 0x0538
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSplitScreenPositionChanged__Delegate; // 0x0550 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LocalPlayer_TA");
}

return uClassPointer;
};

void __LocalPlayer_TA__OnCreated_0x1(class UOnlineLegalText_X* O);

```

```

void ApplyNetworkSettings(class UNetworkSave_TA* NetworkSave);
class USaveData_TA* GetSaveData();
class UObject* GetProfileObject(class UClass* saveClass);
class UObject* GetSaveObject(class UClass* saveClass);
void HandleNewEula(class UOnlineLegalText_X* Eula);
bool JoinServerFromCommandline();
void CheckForRankedReconnect();
void SetOpenedStartMenu(unsigned long bOpen);
void HandleOnlineLoginChanged(class UOnlinePlayerAuthentication_X* Auth);
void UpdateRankedReconnect();
void HandleGameStateChanged(class AGameEvent_TA* GameEvent);
void ClearLoginDelegates();
void OnContinueWithNoUser(class UGFxModal_X* Modal);
void OnShowAccountPicker(class UGFxModal_X* Modal);
void OnLoginCancelled();
void OnLoginFailed(uint8_t LocalUserNum, uint8_t ErrorCode);
void OnControllerPairingChanged(int32_t NewLocalUserNum, int32_t PreviousLocalUserNum);
void OnLoginChange(uint8_t LocalUserNum);
void OnEpicLoginChange(uint8_t LocalUserNum);
void ForceSaveGame();
static bool IsControllerAssociatedWithLoggedInUser(uint8_t LocalUserNum);
bool IsUserLoggedIn();
void UnregisterUserStatusChangeFunctions();
void RegisterUserStatusChangeFunctions();
void PrintDebugInfo(class UDebugDrawer* Drawer);
uint8_t GetSplitScreenPosition(class UGameViewportClient* InViewportClient);
void HandleSplitScreenTypeChanged(class UGameViewportClient_TA* InViewportClient);
void NotifyWhenProfileSet(struct FScriptDelegate Callback);
void SetProfile(class UProfile_TA* NewProfile);
void SetDefaultProfile();
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA* SaveData, class UError* Error);
void HandleLocalPlayerLeave(class ULocalPlayer* Player);
void HandlePreLoadMap(class FString MapName);
void HandleRegisteredController(int32_t InLocalPlayerNum, int32_t InControllerID);
void HandleConnectionQualityConfigChanged();
void OnRemoved();
void OnCreated();
struct FUniqueNetId eventGetUniqueNetId();
void EventSplitScreenPositionChanged(class ULocalPlayer_TA* LocalPlayer);
void EventProfileSet(class ULocalPlayer_TA* LocalPlayer);
};

// Class TAGame.MapData_TA
// 0x0028 (0x0060 - 0x0088)
class UMapData_TA : public UObject
{
public:
class UTexture2D*           Thumbnail;          // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UMapData_TA*           VariantOf;         // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
uint8_t                      WeatherVariant;    // 0x0070 (0x0001)
[0x0000000000000001] (CPF_Edit)

```

```

float RandomWeight; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)

TArray<class FString> AdditionalGameTags; // 0x0078 (0x0010)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapData_TA");
}

return uClassPointer;
};

static float __MapData_TA__PickRandomMap_0x1(float Sum, class UMapData_TA* M);
static class UMapData_TA* GetNewMapData(class FString MapName, float InRandomWeight,
class UMapData_TA* InVariantOf, class UObject* Owner);
static class UMapData_TA* PickPreferredRandomMap(int32_t NumPlayers, float
ForcedIndexRandWeight, float ForcedMapRandWeight, TArray<class UMapData_TA*>& Maps,
TArray<struct FName>& Likes, TArray<struct FName>& Dislikes);
static TArray<class UMapData_TA*> GetMapTypes(TArray<class UMapData_TA*> Maps);
static int32_t GetWeightedIndex(TArray<float> PreferredWeights, float ForcedRandWeight);
static TArray<float> GetPreferredWeights(int32_t NumPlayers, struct FName MapSetName,
TArray<class UMapData_TA*>& Maps, TArray<struct FName>& Likes, TArray<struct FName>&
Dislikes);
static TArray<float> GetWeightsFromPoints(TArray<class UMapData_TA*> Maps,
TArray<int32_t> Points, int32_t NumPlayers, struct FName MapSetName);
static TArray<int32_t> GetPoints(TArray<class UMapData_TA*> Maps, TArray<struct FName>
Likes, TArray<struct FName> Dislikes);
static TArray<class UMapData_TA*> GetNonVariantMaps(TArray<class UMapData_TA*>& Maps);
static class UMapData_TA* PickRandomMap(TArray<class UMapData_TA*> Maps, float
ForcedRandomWeight);
class FString GetLocalizedVariantName();
class FString GetLocalizedBaseName();
class FString eventGetLocalizedName();
};

// Class TAGame.MapDataLoader_TA
// 0x0020 (0x0060 - 0x0080)
class UMapDataLoader_TA : public UObject
{
public:
TArray<struct FMapSelectorPair> MapSelectors; // 0x0060 (0x0010)
[0x000000000480000] (CPF_Component | CPF_NeedCtorLink)
TArray<struct FMapImageLoadRequest> MapImageLoadRequests; // 0x0070 (0x0010) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapDataLoader_TA");
}

return uClassPointer;
};

void eventUpdateAvailableMaps();
void LoadLargeMapImage(class UMapData_TA* Map, struct FScriptDelegate OnLoaded);
class UMapSelector_TA* GetMapSelector(class UMapSet_TA* MapSet);
static class UMapDataLoader_TA* GetInstance();
};

// Class TAGame.MapList_TA
// 0x00A8 (0x0060 - 0x0108)
class UMapList_TA : public UObject
{
public:
TArray<class UMapData_TA*> AllMaps; // 0x0060 (0x0010)
[0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
TArray<class UMapData_TA*> AllMaps_Pristine; // 0x0070
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<class UMapData_TA*> SortedMaps; // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UMapData_TA*> SortedMaps_Pristine; // 0x0090
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<struct FName> AdditionalCookedMaps; // 0x00A0
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FMapGroup> MapGroups; // 0x00B0 (0x0010)
[0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FMapGroup> GeneratedMapGroups; // 0x00C0
(0x0010) [0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
TArray<struct FMapGroup> GeneratedMapGroups_Pristine; // 0x00D0
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
class UMapSet_TA* SelectableMaps; // 0x00E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UMapSet_TA* SeasonMaps; // 0x00E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UMapSet_TA* SeasonPlayoffsMaps; // 0x00F0 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<class UMapSet_TA*> AllMapSets; // 0x00F8 (0x0010)
[0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapList_TA");
}
}

```

```

}

return uClassPointer;
};

TArray<class UMapData_TA*> GetSortedMaps(TArray<class UMapData_TA*> UnsortedMaps);
class UMapData_TA* GetMap(struct FName MapName);
};

// Class TAGame.MapSet_TA
// 0x0060 (0x0060 - 0x00C0)
class UMapSet_TA : public UObject
{
public:
    TArray<class UMapSet_TA*>           IncludeSets;           // 0x0060 (0x0010)
    [0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<class UMapSet_TA*>           ExcludeSets;          // 0x0070 (0x0010)
    [0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<class UMapData_TA*>           IncludeMaps;         // 0x0080 (0x0010)
    [0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<class UMapData_TA*>           ExcludeMaps;        // 0x0090 (0x0010)
    [0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<class UMapData_TA*>           Maps;                // 0x00A0 (0x0010)
    [0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
    TArray<class UMapData_TA*>           Maps_Pristine;       // 0x00B0 (0x0010)
    [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.MapSet_TA");
        }
    }

    return uClassPointer;
};

class UMapData_TA* GetMap(struct FName MapName);
class UMapData_TA* GetRandomMap();
};

// Class TAGame.MatchBroadcast_TA
// 0x0038 (0x0268 - 0x02A0)
class AMatchBroadcast_TA : public AActor
{
public:
    class AGameEvent_Soccar_TA*           GameEvent;           // 0x0268 (0x0008)
    [0x0000000100002020] (CPF_Net | CPF_Transient)
    TArray<class APlayerReplicationInfo*> RecordedPlayers;      // 0x0270
    (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    unsigned long                         bBroadcasting : 1; // 0x0280 (0x0004)
}

```

```
[0x0000000000002000] [0x00000001] (CPF_Transient)
struct FScriptDelegate           __EventLog__Delegate;           // 0x0288 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchBroadcast_TA");
}

return uClassPointer;
};

struct FName __MatchBroadcast_TA__RecordStartEvents_0x1(class UMutator_TA* M);
void eventClientLog(class FString EventJson);
void RecordFunction();
int32_t GetPlayerID(class APlayerReplicationInfo* PRI);
void RecordMatchEnd();
void RecordOvertime();
void RecordPlayerStat(int32_t PlayerID, struct FName Stat);
void RecordPlayerAttackStat(int32_t AttackerID, int32_t VictimID, struct FName Stat, struct FVector AttackerLocation, struct FVector AttackerVelocity, struct FRotator AttackerRotation);
void RecordPlayerHitBallStat(int32_t PlayerID, struct FName Stat, float HitTime, struct FVector CarLocation, struct FVector CarVelocity, struct FRotator CarRotation, struct FVector BallLocation, struct FVector BallVelocity);
void RecordGoalScored(int32_t AttackingTeam, int32_t DefendingTeam, struct FVector BallLocation, struct FVector BallVelocity);
void RecordPlayerScoreHit(int32_t PlayerID, float HitTime, struct FVector CarLocation, struct FVector CarVelocity, struct FRotator CarRotation, struct FVector BallLocation, struct FVector BallVelocity);
void RecordTeamScore(int32_t Team, int32_t Score);
void RecordLeaveTeam(int32_t PlayerID);
void RecordJoinTeam(int32_t PlayerID, int32_t Team);
void RecordLeaveGame(int32_t PlayerID);
void RecordJoinServer(int32_t PlayerID, struct FUniqueNetId PlayerNetId, class FString PlayerName, struct FUniqueNetId PartyLeaderNetID, class FString IP);
void RecordJoinGame(int32_t PlayerID, struct FUniqueNetId PlayerNetId, class FString PlayerName);
void RecordMatchStart(uint64_t UTC, class FString MapName, struct FName GameType, struct FName MatchType, int32_t PlaylistId, TArray<struct FName> Mutators);
float eventGetGameTimeSeconds();
void HandleScoreUpdated(class ATeam_TA* Team);
void RecordInitialScores();
void RecordInitialPlayers();
void RecordStartEvents();
void HandleMatchEnded(class AGameEvent_Soccar_TA* G);
void HandleOvertime(class AGameEvent_Soccar_TA* G);
void HandleStat(class AStatFactory_TA* Factory, class APRI_TA* PRI, class UStatEvent_TA* StatEvent, class ABall_TA* Ball, int32_t BallHitIndex, class APRI_TA* Victim, class UGoal_TA* Goal);
```

```

void HandleGoalScored(class AGameEvent_Soccar_TA* G, class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
void HandleTeamChanged(class APRI_X* PRI);
void HandlePlayerRemoved(class AGameEvent_TA* G, class APRI_TA* PRI);
class FString GetIP(class APRI_TA* PRI);
struct FUniqueNetId GetPartyLeader(struct FUniqueNetId Member);
void HandlePlayerAdded(class AGameEvent_TA* G, class APRI_TA* PRI);
void StopRecording();
void StartRecording();
void HandleStatFactory(class AGameEvent_Soccar_TA* G);
void HandleTeamsCreated(class AGameEvent_Team_TA* G);
void eventDestroyed();
void Start();
void ServerStart();
void Init(class AGameEvent_Soccar_TA* InGameEvent);
void ClientCheckReady();
void eventOnOwnerChanged();
void eventReplicatedEvent(struct FName VarName);
void EventLog(class AMatchBroadcast_TA* Broadcast, class FString EventJson);
};

// Class TAGame.MatchBroadcastLog_TA
// 0x0008 (0x0060 - 0x0068)
class UMatchBroadcastLog_TA : public UObject
{
public:
    struct FPointer           Log;                      // 0x0060 (0x0008)
    [0x0000000000000000] (CPF_Native)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.MatchBroadcastLog_TA");
        }
    }

    return uClassPointer;
};

void Append(class FString EventJson);
void Init(class FString Filename);
};

// Class TAGame.MaterialParamsSet_TA
// 0x0030 (0x0060 - 0x0090)
class UMaterialParamsSet_TA : public UObject
{
public:
    TArray<struct FMaterialTextureParam>      TextureParameters;          // 0x0060
    [0x0010] [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<struct FMaterialVectorParam>         VectorParameters;          // 0x0070
}

```

```

(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FMaterialScalarParam> ScalarParameters; // 0x0080
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MaterialParamsSet_TA");
}

return uClassPointer;
};

void Apply(class UMaterialInstanceConstant* MatInst);

// Class TAGame.MaxActorsGroup_TA
// 0x0040 (0x0070 - 0x00B0)
class UMaxActorsGroup_TA : public UComponent
{
public:
struct FPointer VfTable_FTickableObject; // 0x0070 (0x0008)
[0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
int32_t MaxActors; // 0x0078 (0x0004)
[0x0000000000000003] (CPF>Edit | CPF_Const)
float UpdateDelay; // 0x007C (0x0004)
[0x0000000000000003] (CPF>Edit | CPF_Const)
float NextUpdateTime; // 0x0080 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
TArray<struct FMaxActorCallback> Actors; // 0x0088 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __OnToggledDelegate__Delegate; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MaxActorsGroup_TA");
}

return uClassPointer;
};

class FString GetEnabledActorsString();
void RemoveListener(class AActor* CheckActor, struct FScriptDelegate OnEnabled, struct FScriptDelegate OnDisabled);

```

```

void AddListener(class AActor* CheckActor, struct FScriptDelegate OnEnabled, struct
FScriptDelegate OnDisabled);
void OnToggledDelegate();
};

// Class TAGame.Message_TA
// 0x00A0 (0x0060 - 0x0100)
class UMessage_TA : public UObject
{
public:
struct FName           Type;           // 0x0060 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class UAkSoundCue*     Sound;          // 0x0068 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class UAkSoundCue*     PrimaryPlayerSound; // 0x0070 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class FString          Key;            // 0x0078 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
class FString          Section;         // 0x0088 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
uint8_t                MessageLevel;   // 0x0098 (0x0001)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class FString          LocalizedMessage; // 0x00A0 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<struct FName>  ParamNames;      // 0x00B0 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<class FString> ParamTokens;     // 0x00C0 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
struct FMessagePacket  Packet;          // 0x00D0 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FMessagePacket  DefaultPacket;   // 0x00E8 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Message_TA");
}

return uClassPointer;
};

class FString eventGetObjectTextW(class UObject* O);
void Trigger(class APlayerController_TA* Receiver, struct FMessagePacket InPacket);
struct FMessagePacket GetPacket();
void BroadcastLocal(TArray<class AController*> Receivers);
void Broadcast(TArray<class AController*> Receivers);
void SendTo(class APlayerController_TA* Receiver);
class UMessage_TA* SetInt(struct FName ParamName, int32_t Value);
class UMessage_TA* SetObject(struct FName ParamName, class UObject* Value);

```

```
class UMessage_TA* Set(struct FName ParamName, class FString Value);
class FString GetText(struct FMessagePacket InPacket);
};

// Class TAGame.MutatorGameSetting_TA
// 0x00348 (0x0064 - 0x0098C)
class UMutatorGameSetting_TA : public UGameSetting_X
{
public:
TArray<class FString> RequiresMutators; // 0x0068 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class FString> DisallowedWithMutators; // 0x0078 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString MutatorPackage; // 0x0088 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

int32_t MutatorGroupID; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MutatorGameSetting_TA");
}

return uClassPointer;
};

class UMutator_TA* GetMutator();
class FString eventGetMutatorArchetypePath();
};

// Class TAGame.NavigationHandle_TA
// 0x000C (0x01A8 - 0x01B4)
class UNavigationHandle_TA : public UNavigationHandle
{
public:
struct FVector OldDriveLocation; // 0x01A8 (0x000C)
[0x0000000000002002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NavigationHandle_TA");
}
}
```

```

return uClassPointer;
};

void GetNextDriveDestination(struct FVector CurrentDestination, struct FVector& OutDestination);
bool PathCache_Empty();
};

// Class TAGame.NetworkConfig_TA
// 0x003C (0x0078 - 0x00B4)
class UNetworkConfig_TA : public UOnlineConfig_X
{
public:
    struct FNetworkLimit           ReplicationRate;          // 0x0078 (0x0010)
    [0x0000000000000001] (CPF_Edit)
    struct FNetworkLimit           ClientRate;             // 0x0088 (0x0010)
    [0x0000000000000001] (CPF_Edit)
    struct FNetworkLimit           ClientInputRate;        // 0x0098 (0x0010)
    [0x0000000000000001] (CPF_Edit)
    int32_t                      MaxClientUploadFrames;   // 0x00A8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                 bForceNetPackets : 1;    // 0x00AC (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    float                         ForcePacketSignificantDistance; // 0x00B0 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.NetworkConfig_TA");
        }

        return uClassPointer;
    };

    void UndoNetworkLimit(struct FNetworkLimit& Limit, int32_t& Min, int32_t& Max);
    void ApplyNetworkLimit(struct FNetworkLimit& Limit, int32_t& Min, int32_t& Max);
    void Undo();
    void Apply();
};

// Class TAGame.NetworkInputBuffer_TA
// 0x0060 (0x0268 - 0x02C8)
class ANetworkInputBuffer_TA : public AActor
{
public:
    class UPhysicsConfig_TA*      PhysicsConfig;           // 0x0268 (0x0008)
    [0x0000800000000000]
    class UNetworkJitterSettings_TA* JitterSettings;       // 0x0270 (0x0008)
};

```

```

[0x0000800000000000]
class UEngineShare_TA*           EngineShare;          // 0x0278 (0x0008)
[0x0000800000000000]
class UClientInputData_TA*       PrimaryClient;        // 0x0280 (0x0008)
[0x0000000000000000]
TArray<class UClientInputData_TA*> Clients;           // 0x0288 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
float                           TimeSinceLastMovePacket; // 0x0298 (0x0004)
[0x0000000000000000]
int32_t                         MaxInputFramesPerPacket; // 0x029C (0x0004)
[0x0000000000000000]
TArray<float>                  PingTimestamps;        // 0x02A0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
float                           LastTickTime;          // 0x02B0 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FInputBufferStats         Stats;               // 0x02B4 (0x0008)
[0x0000000000000000]
class UNetworkPingEvent_X*      PingEvent;           // 0x02C0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NetworkInputBuffer_TA");
}

return uClassPointer;
};

void ResetInputBuffer();
void ServerResetInputBuffer();
void UpdatePing();
void ClientAckFrame(float TimeStamp, int32_t InClientFrame, struct FInputBufferStats InStats);
void ResetStats();
void SendClientAck(float TimeStamp);
void eventServerTick();
void TrimInputFrames(class UClientInputData_TA* Client);
void ReceiveClientFrames(class UClientInputData_TA* Client, struct FClientPacketData& Packet);
void ReceiveNewClientPacket(class UClientInputData_TA* Client, struct FClientPacketData& Packet);
void ReceiveClientPacket(class UClientInputData_TA* Client, struct FClientPacketData& Packet);
void ServerReceivePacket(class APlayerController_TA* PC, struct FClientPacketData Packet);
void ClientSendPacket(class UClientInputData_TA* Client);
void SendClientMovePackets();
bool ShouldSendClientMovePackets();
void eventClientTick();
void RemoveClient(class APlayerController_TA* PC);
class UClientInputData_TA* AddClient(class APlayerController_TA* PC);
class UClientInputData_TA* GetClient(class APlayerController_TA* PC);
void eventDestroyed();

```

```

void eventPostBeginPlay();
};

// Class TAGame.NetworkInputBuffer_CSTS_TA
// 0x0030 (0x02C8 - 0x02F8)
class ANetworkInputBuffer_CSTS_TA : public ANetworkInputBuffer_TA
{
public:
    class UPhysicsConfig_TA*           Config;           // 0x02C8 (0x0008)
    [0x0000800000000000]
    float                             LastPacketReceiveTime; // 0x02D0 (0x0004)
    float                             CurrentTimeDilation; // 0x02D4 (0x0004)
    int32_t                           SequentialMissedInputs; // 0x02D8 (0x0004)
    int32_t                           BufferLowWatermark; // 0x02DC (0x0004)
    TArray<int32_t>                  BufferLowHistory; // 0x02E0 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)
    int32_t                           BufferLowHistoryIndex; // 0x02F0 (0x0004)
    int32_t                           ExtraBuffer; // 0x02F4 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.NetworkInputBuffer_CSTS_TA");
        }

        return uClassPointer;
    }

    void ClientAckFrame_CSTS(float TimeStamp, int32_t InClientFrame, struct FInputBufferStats
    InStats, uint8_t SatLevel);
    uint8_t GetInputBufferSaturationLevel(int32_t InputBufferSize);
    void SendClientAck(float TimeStamp);
    void UpdateExtraBuffer();
    void StartUpdatingExtraBuffer();
    void ResetInputBuffer();
    void ReceiveNewClientPacket(class UClientInputData_TA* Client, struct FClientPacketData&
    Packet);
    void eventPostBeginPlay();
};

// Class TAGame.NetworkInputBuffer_ServerConsume_TA
// 0x000C (0x02C8 - 0x02D4)
class ANetworkInputBuffer_ServerConsume_TA : public ANetworkInputBuffer_TA
{

```

```

public:
class UClientJitterBuffer_TA*           JitterBuffer;          // 0x02C8 (0x0008)
[0x0000000000000000]
unsigned long                          bFillingBuffer : 1;    // 0x02D0 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NetworkInputBuffer_ServerConsume_TA");
}

return uClassPointer;
};

void ResetInputBuffer();
void ReceiveNewClientPacket(class UClientInputData_TA* Client, struct FClientPacketData&
Packet);
void eventPostBeginPlay();
};

// Class TAGame.NetworkInputBuffer_STS_TA
// 0x0030 (0x02C8 - 0x02F8)
class ANetworkInputBuffer_STS_TA : public ANetworkInputBuffer_TA
{
public:
class UClientJitterBuffer_TA*           JitterBuffer;          // 0x02C8 (0x0008)
[0x0000000000000000]
float                                 ExtraSimTimeRemaining; // 0x02D0 (0x0004)
[0x0000000000000000]
struct FSimTimeAdjustDebugData        Debug;                // 0x02D4 (0x0008)
[0x0000000000000000]
float                                 TimeLastSimTimeAdjustment; // 0x02DC (0x0004)
[0x0000000000000000]
float                                 TimeClientAckdAdjustSimTime; // 0x02E0 (0x0004)
[0x0000000000000000]
uint8_t                               ReplicatedInputBufferSize; // 0x02E4 (0x0001)
[0x0000000100000020] (CPF_Net)
unsigned long                         bAdjustingSimTime : 1;    // 0x02E8 (0x0004)
[0x0000000000000000] [0x00000001]
int32_t                               BufferSizeLowWatermark; // 0x02EC (0x0004)
[0x0000000000000000]
float                                 TimeLastMissedInput;    // 0x02F0 (0x0004)
[0x0000000000000000]
float                                 TimeLastTightenUpBufferCheck; // 0x02F4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NetworkInputBuffer_STS_TA");
}

return uClassPointer;
};

void eventServerFinishedSimTimeAdjust();
void eventClientAdjustSimTime(int32_t FrameAdjustment);
void NotifyClientAdjustSimTime(int32_t FrameAdjustment);
bool CanTellClientToAdjustSimTime();
bool CanTightenUpInputBufferSize();
void eventServerTick();
void BufferSizeChanged(int32_t PrevBufferSize, int32_t BufferSize);
void ReceiveNewClientPacket(class UClientInputData_TA* Client, struct FClientPacketData& Packet);
void eventReplicatedEvent(struct FName VarName);
void eventPostBeginPlay();
};

// Class TAGame.NetworkJitterSettings_TA
// 0x001C (0x0060 - 0x007C)
class UNetworkJitterSettings_TA : public UObject
{
public:
float MinTimeBetweenPackets; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxTimeBetweenPackets; // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BufferStdDeviations; // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinBufferTime; // 0x006C (0x0004)
[0x0000000000000001] (CPF_Edit)
float RollingAverageTime; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float InputBufferTightenUpSizeFrequency; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t MaxSingleAdjustmentFrames; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NetworkJitterSettings_TA");
}

return uClassPointer;
}

```

```

};

};

// Class TAGame.Note_Linked_TA
// 0x0010 (0x0278 - 0x0288)
class ANote_Linked_TA : public ANote
{
public:
class ANote_Linked_TA* [0x0000000000000001] (CPF_Edit) NextNote; // 0x0278 (0x0008)
class ANote_Linked_TA* [0x0000000000000001] (CPF_Edit) PrevNote; // 0x0280 (0x0008)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Note_Linked_TA");
}

return uClassPointer;
};

};

// Class TAGame.Notification_TA
// 0x0110 (0x0060 - 0x0170)
class UNotification_TA : public UObject
{
public:
int32_t NotificationID; // 0x0060 (0x0004)
[0x000000040000000] (CPF>EditInlineNotify)
class FString Title; // 0x0068 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString Body; // 0x0078 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
unsigned long bPopUp : 1; // 0x0088 (0x0004)
[0x000000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long bRemoveAfterDisplayed : 1; // 0x0088 (0x0004)
[0x000000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long bPopUpShown : 1; // 0x0088 (0x0004)
[0x000000040000000] [0x00000004] (CPF>EditInlineNotify)
unsigned long bSave : 1; // 0x0088 (0x0004)
[0x0000004000000002] [0x00000008] (CPF_Const)
unsigned long bPauseWhenModalsActive : 1; // 0x0088 (0x0004)
[0x0000004000000000] [0x00000010]
float PopUpDuration; // 0x008C (0x0004)
[0x000000040000000] (CPF>EditInlineNotify)
uint64_t Expiration; // 0x0090 (0x0008)
[0x0000000000000000]

```

```
class UNotificationManager_TA*          NotificationManager;           // 0x0098
(0x0008) [0x0000000000002000] (CPF_Transient)
class UClass*                          GFxDataClass;                // 0x00A0 (0x0008)
[0x0000004000000002] (CPF_Const)
class UCrossplayConfig_X*             CrossplayConfig;            // 0x00A8 (0x0008)
[0x000800000000000]
struct FScriptDelegate               AcceptCallBack;              // 0x00B0 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate               DenyCallBack;                // 0x00C8 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate               ShownCallBack;              // 0x00E0 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate               HiddenCallBack;             // 0x00F8 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate               __ShownDelegate__Delegate; // 0x0110
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate               __HiddenDelegate__Delegate; // 0x0128
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate               __ClickDelegate__Delegate; // 0x0140 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate               __EventRemoveNotification__Delegate; // 0x0158
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Notification_TA");
}

return uClassPointer;
};
```

```
bool IsCrossplayNotificationAllowed(struct FUniqueNetId OtherPlayerID);
bool ShouldShow();
struct FUniqueNetId GetSenderID();
class UNotification_TA* SetHiddenDelegate(struct FScriptDelegate Callback);
class UNotification_TA* SetShownDelegate(struct FScriptDelegate Callback);
class UNotification_TA* SetDenyDelegate(struct FScriptDelegate Callback);
class UNotification_TA* SetAcceptDelegate(struct FScriptDelegate Callback);
void RemoveNotification();
void HandleNotificationSaveAdded(class UNotificationSave_TA* NotificationSave);
void eventSaveNotification();
class ULocalPlayer_TA* GetLocalPlayer();
void OnPopUpDisplayed();
void SetShown(unsigned long blnShown);
class UNotification_TA* SetBody(class FString InBody);
class UNotification_TA* SetTitle(class FString InTitle);
void eventOnPopUpOnlyNotificationCreated();
void eventOnNotificationCreated(unsigned long bSuppressPopup);
void OnHidden();
```

```

void OnShown();
void ClickDeny();
void ClickAccept();
bool IsExpired();
void Expire();
void UpdateExpiration();
void SetExpiration(uint64_t InExpiration);
void eventInit(class UNotificationManager_TA* InNotificationManager);
void EventRemoveNotification(class UNotification_TA* Notification);
void ClickDelegate(class UNotification_TA* Notification);
void HiddenDelegate(class UNotification_TA* Notification);
void ShownDelegate(class UNotification_TA* Notification);
};

// Class TAGame.GenericNotification_TA
// 0x0000 (0x0170 - 0x0170)
class UGenericNotification_TA : public UNotification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GenericNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.NotificationManager_TA
// 0x0078 (0x0060 - 0x00D8)
class UNotificationManager_TA : public UObject
{
public:
TArray<class UNotification_TA*> Notifications; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t CurrentNotificationID; // 0x0070 (0x0004)
[0x0000000000000000]
struct FScriptDelegate __EventNotificationAdded__Delegate; // 0x0078
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventNotificationRemoved__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __HasExistingNotificationDelegate__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __NotificationFilterFunction__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NotificationManager_TA");
}

return uClassPointer;
};

int32_t TypeCount(class UClass* NotificationClass);
void RemoveNotificationsOfType(class UClass* NotificationClass);
void RemoveNotification(class UNotification_TA* Notification);
void RemoveAllNotificationsFromSender(struct FUniqueNetId SenderId);
class UNotification_TA* PopUpOnlyNotification(class UClass* NotificationClass);
class UNotification_TA* AddNotification(class UClass* NotificationClass, unsigned long bSuppressPopup);
int32_t GetNewNotificationID();
bool HasExistingNotification(struct FScriptDelegate HasExistingDelegate);
void RemoveNotificationTypeFromSender(class UClass* NotificationClass, struct FUniqueNetId SenderId);
void HandleChallengesSynced(class UChallengeDefaultManager_TA* DefaultManager);
void HandleNotificationSaveAdded(class UNotificationSave_TA* NotificationSave);
class ULocalPlayer_TA* GetLocalPlayer();
void eventConstruct();
bool NotificationFilterFunction(class UNotification_TA* InNotification);
bool HasExistingNotificationDelegate(class UNotification_TA* OtherNotification);
void EventNotificationRemoved(class UNotificationManager_TA* NotificationManager, class UNotification_TA* Notification);
void EventNotificationAdded(class UNotificationManager_TA* NotificationManager, class UNotification_TA* Notification);
};

// Class TAGame.OnlineProduct_TA
// 0x0070 (0x0060 - 0x00D0)
class UOnlineProduct_TA : public UObject
{
public:
int32_t ProductID; // 0x0060 (0x0004)
[0x0000004000000000]
struct FProductInstanceID InstanceID; // 0x0068 (0x0010)
[0x0000004000000000]
TArray<class UProductAttribute_TA*> Attributes; // 0x0078 (0x0010)
[0x0000008000400000] (CPF_NeedCtorLink)
int32_t SeriesID; // 0x0088 (0x0004)
[0x0000004000000000]
int32_t TradeHold; // 0x008C (0x0004)
[0x0000004000000000]
uint64_t AddedTimestamp; // 0x0090 (0x0008)
[0x0000004000000000]
class FString CachedSortLabel; // 0x0098 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)

```

```
class FString           CachedShortSortLabel;          // 0x00A8 (0x0010)
[0x00000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
struct FProductHashID CachedHash;                  // 0x00B8 (0x0004)
[0x00000000000002002] (CPF_Const | CPF_Transient)
class FString           CachedLocalizedSeries;        // 0x00C0 (0x0010)
[0x00000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineProduct_TA");
}

return uClassPointer;
};

static struct FOnlineProductAttribute
__OnlineProduct_TA__InstanceTempOnlineProduct_0x1(class UProductAttribute_TA* A);
bool __OnlineProduct_TA__RequiresDynamicThumbnail_0x1(class UProductAttribute_TA* Att);
void __OnlineProduct_TA__OnLoaded_0x3(class UProductAttribute_Certified_TA* Certified);
bool __OnlineProduct_TA__OnLoaded_0x2(class UProductAttribute_Certified_TA* Certified);
void __OnlineProduct_TA__OnLoaded_0x1(class UProductAttribute_TA* A);
static int32_t __OnlineProduct_TA__SortOnlineProductsByQuality_0x2(class UOnlineProduct_TA*
L, class UOnlineProduct_TA* R);
void SetTradehold(int32_t InTradehold);
static TArray<class UOnlineProduct_TA*> SortOnlineProductsByQuality(TArray<class
UOnlineProduct_TA*>& Products);
static int32_t SortOnlineProductsBySlot(class UOnlineProduct_TA* P0, class
UOnlineProduct_TA* P1);
void OnLoaded();
void RemoveAttributes(class UClass* AttributeClass);
void RemoveAttribute(class UProductAttribute_TA* InAttribute);
void AddAttribute(class UProductAttribute_TA* InAttribute);
void SetProductID(int32_t InProductID);
bool GetIsUntradeable();
static void SortByTradeHoldAscending(TArray<class UOnlineProduct_TA*>& Products);
static void SortByTradeHoldDescending(TArray<class UOnlineProduct_TA*>& Products);
bool CanBeArchived();
bool RequiresDynamicThumbnail();
class FString GetDebugString();
bool HasAttributesOtherThanGiven(TArray<class UClass*>& AttributeClasses);
class FString GetAttributeString(TArray<class UClass*> AttributesToIgnore);
struct FProductHashID OnlineProductHash();
struct FProductHashID OnlineProductHashV1();
void GetPaintedColorAttribute(class FString& ProductPaintColorName, struct FColor&
ProductPaintColor);
uint8_t GetQuality();
TArray<class UProductAttribute_TA*> GetReplicatedAttributes();
TArray<class UProductAttribute_TA*> GetAttributes();
class FString GetLocalizedSeries();
```

```

class FString GetLongLabel();
int32_t GetBlueprintSeriesID();
uint8_t GetBlueprintType();
bool IsBlueprint();
class UProduct_TA* GetProduct();
class UProductAttribute_TA* GetAttribute(class UClass* AttributeClass);
struct FOnlineProductData InstanceOnlineProductData();
static class UOnlineProduct_TA* InstanceTempOnlineProduct(struct FProductHashID HashID);
static class UOnlineProduct_TA* CreateOnlineProduct(struct FOnlineProductData& ProductData);
static void ProcessOnlineData(TArray<class UOnlineProduct_TA*>& OutProducts, TArray<struct FOnlineProductData>& InProductData);
};

// Class TAGame.OnlineProductStore_TA
// 0x0020 (0x0070 - 0x0090)
class UOnlineProductStore_TA : public UComponent
{
public:
TArray<class UOnlineProductStoreSet_TA*> Sets; // 0x0070
(0x0010) [0x0000004004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink |
CPF_EditInline)
struct FProductInstanceId CurrentTempInstanceId; // 0x0080
(0x0010) [0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineProductStore_TA");
}

return uClassPointer;
};

bool __OnlineProductStore_TA__HasAnyProducts_0x1(class UOnlineProductStoreSet_TA* S);
bool HasAnyProducts();
struct FProductInstanceId GetTempInstanceId();
int32_t GetProductIDFromOnlinelD(struct FProductInstanceId InstanceID);
TArray<class UOnlineProduct_TA*> FindProductsFromHash(struct FProductHashID HashID);
class UOnlineProduct_TA* FindProductFromHash(struct FProductHashID HashID);
class UOnlineProduct_TA* FindProduct(struct FProductInstanceId InstanceID);
void RemoveSet(class UOnlineProductStoreSet_TA* Set);
class UOnlineProductStoreSet_TA* CreateSet();
static class UOnlineProductStore_TA* GetInstance();
void eventConstruct();
};

// Class TAGame.OnlineSaveRecord_TA
// 0x0028 (0x0060 - 0x0088)
class UOnlineSaveRecord_TA : public UObject
{

```

```

public:
int32_t Version; // 0x0060 (0x0004)
[0x0000000000000000] TimeStamp; // 0x0068 (0x0008)
uint64_t SaveTick; // 0x0070 (0x0004)
[0x0000000000000000] TArray<struct FOnlineSaveObject> SaveObjects; // 0x0078 (0x0010)
[0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineSaveRecord_TA");
}

return uClassPointer;
};

static bool DeserializeRecord(class UOnlineSaveRecord_TA* Record, class UObject* SaveObjectOuter, TArray<uint8_t*>& FileContents);
static bool SerializeRecord(class UOnlineSaveRecord_TA* Record, TArray<uint8_t*>& FileContents);
};

// Class TAGame.PaintDatabase_TA
// 0x0064 (0x0060 - 0x00C4)
class UPaintDatabase_TA : public UObject
{
public:
TArray<class UProductPaint_TA*> Paints; // 0x0060 (0x0010)
[0x000000000420003] (CPF>Edit | CPF_Const | CPF_EditConst | CPF_NeedCtorLink)
struct FMap_Mirror PaintNameToPaintID; // 0x0070 (0x0050)
[0x0000000000001002] (CPF_Const | CPF_Native)
unsigned long bClickToRebuild : 1; // 0x00C0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF>Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PaintDatabase_TA");
}

return uClassPointer;
};

```

```

static void ComputeHSLDifferenceBetweenTwoPaints(int32_t DestinationPaintID, int32_t
SourcePaintID, float& Hue, float& Saturation, float& Lightness);
class UProductPaint_TA* GetPaintByName(struct FName PaintName);
class UProductPaint_TA* GetPaint(int32_t Id);
struct FName GetPaintName(int32_t PaintID);
int32_t GetPaintID(struct FName PaintName);
};

// Class TAGame.ParticleSystemComponent_TA
// 0x0018 (0x03E0 - 0x03F8)
class UParticleSystemComponent_TA : public UParticleSystemComponent
{
public:
struct FVector PreviewFakeVelocity; // 0x03E0 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector PodiumFakeVelocity; // 0x03EC (0x000C)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ParticleSystemComponent_TA");
}

return uClassPointer;
};

static void AttachToWheel(class UParticleSystemComponent* PSC, class
USkeletalMeshComponent* WheelMesh, class UProductAsset_Wheel_TA* WheelAsset, struct
FName Socket, TArray<class UProductAttribute_TA*>& AttributesToApply);
};

// Class TAGame.Persona_TA
// 0x0168 (0x0060 - 0x01C8)
class UPersona_TA : public UObject
{
public:
class FString PlayerName; // 0x0060 (0x0010)
[0x0000004040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString UnSanitizedPlayerName; // 0x0070 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
struct FUniqueNetId PlayerID; // 0x0080 (0x0048)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint8_t OnlinePresenceState; // 0x00C8 (0x0001)
[0x0000004040000000] (CPF_EditInlineNotify)
uint8_t OnlinePresenceState_Platform; // 0x00C9 (0x0001)
[0x0000004000000000]
uint8_t OnlinePresenceState_PsyNet; // 0x00CA (0x0001)
[0x0000004000000000]
uint8_t OnlinePresenceState_Epic; // 0x00CB (0x0001)

```

```

[0x0000004000000000]
uint8_t Platform; // 0x00CC (0x0001)
[0x000000040000000] (CPF_EditInlineNotify)
uint8_t SecondaryPlatform; // 0x00CD (0x0001)
[0x000000040000000] (CPF_EditInlineNotify)
class FString OnlinePresenceMessage; // 0x00D0 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bBlockedStatus : 1; // 0x00E0 (0x0004)
[0x000000404000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bPlatformFriend : 1; // 0x00E0 (0x0004)
[0x000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bPsyNetFriend : 1; // 0x00E0 (0x0004)
[0x000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long bEpicFriend : 1; // 0x00E0 (0x0004)
[0x000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bNoEpicAccountIdFound : 1; // 0x00E0 (0x0004)
[0x000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long bNameSanitized : 1; // 0x00E0 (0x0004)
[0x000000404000000] [0x00000020] (CPF_EditInlineNotify)
unsigned long bCanShowAvatar : 1; // 0x00E0 (0x0004)
[0x000000404000000] [0x00000040] (CPF_EditInlineNotify)
unsigned long bAvatarPermissionDownloaded : 1; // 0x00E0
(0x0004) [0x0000000000000000] [0x00000080]
unsigned long bHidden : 1; // 0x00E0 (0x0004)
[0x000000404000000] [0x00000100] (CPF_EditInlineNotify)
int32_t Presenceld; // 0x00E4 (0x0004)
[0x000000404000000] (CPF_EditInlineNotify)
uint64_t LastActive; // 0x00E8 (0x0008)
[0x000000040000000] (CPF_EditInlineNotify)
class FString PlayerNote; // 0x00F0 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ClubTag; // 0x0100 (0x0010)
[0x000100004040000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString UnsanitizedEpicName; // 0x0110 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
class FString SecondaryPlayerName; // 0x0120 (0x0010)
[0x000000404040000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString UnsanitizedSecondaryPlayerName; // 0x0130
(0x0010) [0x000000400040000] (CPF_NeedCtorLink)
class FString EpicAccountId; // 0x0140 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventUpdated__Delegate; // 0x0150
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventNameSanitized__Delegate; // 0x0168
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventBlockStatusChanged__Delegate; // 0x0180
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventEpicAccountIdSet__Delegate; // 0x0198
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventHiddenChanged__Delegate; // 0x01B0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.Persona_TA");  
}  
  
return uClassPointer;  
};  
  
void __Persona_TA__SetSecondaryPlayerName_0x1(class FString Original, class FString  
Sanitized);  
void SetHidden(unsigned long bInHidden);  
void SetCanShowAvatar(unsigned long bInCanShowAvatar);  
void SetEpicAccountId(class FString InEpicAccountId);  
void SetBlockStatus(unsigned long bBlocked);  
void NotifyOnNameSanitized(struct FScriptDelegate Callback);  
void SetPlayerNameSanitized(class FString Original, class FString Sanitized);  
void SetPlayerNameInternal(class FString NewName, unsigned long bPrimaryName, struct  
FScriptDelegate Callback, unsigned long bEpicName, class FString& OutName, class FString&  
OutUnsanitizedName);  
void SetEpicName(class FString EpicName);  
void SetSecondaryPlayerName(class FString NewName);  
void SetPlayerName(class FString NewName);  
bool CanSendPartyInvite();  
void SetPresenceMessage(class FString PresenceInfo);  
bool IsFriendInGame();  
bool IsEpicFriendInGame();  
bool IsPurePsyNetFriendInGame();  
bool IsPlatformFriendInGame();  
bool IsSamePlatformPsyNetFriendsAllowed();  
bool IsFriend();  
void UpdateDisplayPresence();  
void SetPresenceld(int32_t InPresenceld, unsigned long bHiddenGroup);  
void SetFriendGroupIDFromPresenceState();  
void SetOnlinePresenceState(uint8_t InOnlinePresenceState, uint8_t Origin);  
void SetEpicOnlinePresenceState(uint8_t InOnlinePresenceState, uint8_t InOrigin, class FString  
InPresenceInfo);  
void SetEpicPresenceMessage(class FString PresenceInfo);  
void SetPlatformPresenceMessage(class FString PresenceInfo);  
bool IsCrossPlatform();  
class UPersona_TA* eventUpdateFromOnlineFriend(struct FOnlineFriend OnlineInfo, uint8_t  
Origin);  
void EventHiddenChanged(class UPersona_TA* Persona);  
void EventEpicAccountIdSet(class UPersona_TA* Persona);  
void EventBlockStatusChanged(class UPersona_TA* Persona);  
void EventNameSanitized(class UPersona_TA* Persona);  
void EventUpdated(class UPersona_TA* Persona);  
};  
  
// Class TAGame.Personas_TA  
// 0x00C8 (0x0060 - 0x0128)  
class UPersonas_TA : public UObject
```

```

{
public:
struct FMap_Mirror           PersonaMap;           // 0x0060 (0x0050)
[0x00000000000000001000] (CPF_Native)
unsigned long                 bVerbosePresenceSort : 1;      // 0x00B0 (0x0004)
[0x00000000000000004002] [0x00000001] (CPF_Const | CPF_Config)
class FString                DeletedAccountDisplayName; // 0x00B8 (0x0010)
[0x000000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
TArray<struct FUniqueNetId>   EpicIdRequests;        // 0x00C8 (0x0010)
[0x000000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UOnlineGame_X*          OnlineGame;           // 0x00D8 (0x0008)
[0x0000800000000000]
struct FScriptDelegate         __EventAdded__Delegate; // 0x00E0 (0x0018)
[0x00000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventEpicIdsReceived__Delegate; // 0x00F8
(0x0018) [0x00000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventPersonaHiddenChanged__Delegate; // 0x0110
(0x0018) [0x00000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Personas_TA");
}

return uClassPointer;
};

static bool ShouldCombineFriends(struct FOnlineFriend& PlatformFriend, struct FOnlineFriend& EpicFriend);
class UPersona_TA* CreatePersonaFromOnlineFriend(uint8_t Origin, struct FOnlineFriend& FriendData);
class UPersona_TA* CreatePersona(struct FUniqueNetId PlayerID, class FString PlayerName, uint8_t Origin);
void RequestEpicAccountIds(TArray<struct FUniqueNetId>& RequestedIds);
void HandleOnReceivedLinkedAccounts(unsigned long bSuccess, TArray<struct FLinkedAccountData> LinkedData);
void SendEpicIdRequests(class UPsyNetConnection_X* C);
void UpdatePersonaEpicIds();
class FString GetPlayerEpicAccountId(struct FUniqueNetId PlayerID);
void HandlePersonaHiddenChanged(class UPersona_TA* Persona);
class UPersona_TA* AddPersona(class UPersona_TA* Persona);
class UPersona_TA* eventUpdatePersonaFromData(uint8_t Origin, struct FOnlineFriend& FriendData);
void InsertSortedPersonaData(struct FPersonaDataId Persona, TArray<struct FPersonaDataId>& PersonaData);
void SortPersonaDataAlphabetically(TArray<struct FPersonaDataId>& OutPersonaData);
void SortPersonaData(TArray<struct FPersonaDataId>& OutPersonaData);
void GetAllPersonasInGroup(int32_t Presenceld, TArray<class UPersona_TA*>& OutPersonas);
void AddToMap(class UPersona_TA* Persona);
}

```

```

class UPersona_TA* GetPlatformPersonaFromEpicID(class UOnlinePlayerFriends_X*
OnlineFriends, struct FUniqueNetId EpicId);
class UPersona_TA* GetOrCreateEpicPersonaFromPlatformID(class UOnlinePlayerFriends_X*
OnlineFriends, struct FUniqueNetId PlatformId, class FString PlayerName, uint8_t Origin);
class UPersona_TA* GetEpicPersonaFromPlatformID(class UOnlinePlayerFriends_X*
OnlineFriends, struct FUniqueNetId PlatformId);
struct FUniqueNetId QueryLinkedEpicId(struct FUniqueNetId PlatformId, class
UOnlinePlayerFriends_X* OnlineFriends);
struct FUniqueNetId GetEpicIDFromPlatformID(struct FUniqueNetId PlatformId);
class UPersona_TA* GetPersonaFromID(struct FUniqueNetId PlayerID);
static void ShowBlockedInteractionError(class UPersona_TA* BlockedPersona, class
UGFxShell_TA* Shell);
class UPersona_TA* GetOrCreateFromOnlineFriend(uint8_t Origin, struct FOnlineFriend&
FriendData);
class UPersona_TA* UpdatePersonaName(struct FUniqueNetId PlayerID, class FString
PlayerName);
class UPersona_TA* GetOrCreate(struct FUniqueNetId PlayerID, class FString PlayerName,
uint8_t Origin);
void CreateDefaultPersonas();
void EventPersonaHiddenChanged(class UPersona_TA* Persona);
void EventEpicIdsReceived(unsigned long bSuccess, TArray<struct FLinkedAccountData>
LinkedData);
void EventAdded(class UPersonas_TA* PersonasManager, class UPersona_TA* Persona);
};

// Class TAGame.PhysicalMaterialProperty_TA
// 0x0010 (0x0060 - 0x0070)
class UPhysicalMaterialProperty_TA : public UPhysicalMaterialPropertyBase
{
public:
    struct FName           AkSwitchValue;          // 0x0060 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    unsigned long           bStickyWheels : 1;       // 0x0068 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long           bConsiderForGround : 1;   // 0x0068 (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    float                  GroundToleranceZ;        // 0x006C (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.PhysicalMaterialProperty_TA");
        }

        return uClassPointer;
    };

};

```

```

// Class TAGame.PhysicsConfig_TA
// 0x0178 (0x0078 - 0x01F0)
class UPhysicsConfig_TA : public UOnlineConfig_X
{
public:
TArray<class FString> EnabledFeatures; // 0x0078 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class FString> DisabledFeatures; // 0x0088 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FCarInteractionConfig CarInteractionSettings; // 0x0098 (0x0108)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FSuperSonicConfig SuperSonicSettings; // 0x01A0 (0x000C)
[0x0000000000000000]
unsigned long bUseAckermannSteering : 1; // 0x01AC (0x0004)
[0x0000000000000001]
unsigned long bPredictContacts : 1; // 0x01AC (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long bQuantizePhysics : 1; // 0x01AC (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long bOverrideServerPhysicsState : 1; // 0x01AC (0x0004)
[0x0000000000000000] [0x00000008]
unsigned long bBackupVehicleInputs : 1; // 0x01AC (0x0004)
[0x0000000000000000] [0x00000010]
unsigned long bUseNEQ : 1; // 0x01AC (0x0004)
[0x0000000000000000] [0x00000020]
unsigned long bExtrapolateRendering : 1; // 0x01AC (0x0004)
[0x0000000000000001] [0x00000040] (CPF_Edit)
unsigned long bEnableCustomExplosionPhysics : 1; // 0x01AC
(0x0004) [0x0000000000000000] [0x00000080]
unsigned long bLegacyDropshotTileCollision : 1; // 0x01AC (0x0004)
[0x0000000000000000] [0x00000100]
float JumpLeaveGroundTime; // 0x01B0 (0x0004)
[0x0000000000000000]
struct FSimTimeScaleSettings STS; // 0x01B4 (0x0004)
[0x0000000000000000]
struct FContinuousSimTimeScaleSettings CSTS; // 0x01B8
(0x0034) [0x0000000000000000]
float PushForceConstantScale; // 0x01EC (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PhysicsConfig_TA");
}

return uClassPointer;
};

void DisableFeature(class FString Feature);

```

```

void EnableFeature(class FString Feature);
void Reset();
void Apply();
};

// Class TAGame.PitchTekDrawingComponent_TA
// 0x0033 (0x009D - 0x00D0)
class UPitchTekDrawingComponent_TA : public UActorComponent
{
public:
struct FVector PreviousOwnerPosition; // 0x00A0 (0x000C)
[0x0000000000000000] (CPF_Transient)
uint8_t DecalType; // 0x00AC (0x0001)
[0x0000000000000000]
TArray<struct FWheelContactData> WheelContacts; // 0x00B0
(0x0010) [0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
struct FPitchTekCarBodyContactData CarBodyContactData; // 0x00C0
(0x0010) [0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PitchTekDrawingComponent_TA");
}

return uClassPointer;
};

void QueueJumpBlastDecal(struct FVector JumpBlastLocation);
void QueueDemolitionExplosionDecal(struct FVector DemolitionExplosionLocation);
void QueueGoalExplosionDecal(struct FVector GoalExplosionLocation);
};

// Class TAGame.PlayerVanity_TA
// 0x0128 (0x0070 - 0x0198)
class UPlayerVanity_TA : public UComponent
{
public:
class FString MovieReference; // 0x0070 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture* ToPlayer; // 0x0080 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture2D* PaintableLayer; // 0x0088 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture2D* TintableLayer; // 0x0090 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
unsigned long bColorable : 1; // 0x0098 (0x0004)
[0x0000000040002000] [0x00000001] (CPF_Transient | CPF_EditInlineNotify)
unsigned long bIsDirty : 1; // 0x0098 (0x0004)
[0x0000000000000000] [0x00000002] (CPF_Transient)

```

```

int32_t ColorID; // 0x009C (0x0004)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
struct FColor ColorValue; // 0x00A0 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
struct FColor DefaultValue; // 0x00A4 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
struct FColor PaintColor; // 0x00A8 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class FString InstanceID; // 0x00B0 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t ProductID; // 0x00C0 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
struct FProductHashID HashID; // 0x00C4 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
struct FColor RawDefaultValue; // 0x00C8 (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<class UProductAttribute_TA*> Attributes; // 0x00D0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString PlayerName; // 0x00E0 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
struct FUniqueNetId PlayerID; // 0x00F0 (0x0048)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
class UProfile_TA* Profile; // 0x0138 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UPlayerVanitySave_TA* VanitySaveObject; // 0x0140 (0x0008)
[0x0000000000002000] (CPF_Transient)
uint8_t VanityType; // 0x0148 (0x0001)
[0x0000000000000002] (CPF_Const)
struct FScriptDelegate OnLoadCompleteCallback; // 0x0150
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventOnLoadComplete__Delegate; // 0x0168
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventEquipProduct__Delegate; // 0x0180
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerVanity_TA");
}

return uClassPointer;
};

void __PlayerVanity_TA__InitWithProfile_0x1(class USaveData_TA* _);
bool HasPlayerID(struct FUniqueNetId Id);
void ClearVanity();
void ApplyColor(struct FColor InColor);
int32_t GetDefaultColor(struct FProductInstanceID ProductInstanceId);
void EquipColor(int32_t InColorID);

```

```

void EquipProduct_Internal(struct FProductInstanceID InInstanceId, int32_t InProductID, int32_t
InSelectedColorValue);
void HandleEquippedProductLoaded(struct FAssetLoadResult Result, struct FProductInstanceID
InInstanceId, int32_t InProductID);
void EquipProduct(class UProfile_TA* InProfile, struct FProductHashID InHashID);
void UpdateColorFromProfile(class UProfile_TA* InProfile, int32_t InProductID, int32_t InColorID);
bool IsEmpty();
void UpdateFromAsset(class UProductAsset_PlayerVanity_TA* Asset);
void HandleLoaded(struct FAssetLoadResult Result);
class UClass* GetSaveType();
void SetPlayerID(struct FUniqueNetId InPlayerId);
void UpdateSaveObject();
void Init(int32_t InProductID, struct FProductInstanceID InInstanceId, struct FProductHashID
InHashID, TArray<class UProductAttribute_TA*> InAttributes, struct FScriptDelegate Callback);
void InitWithPlayerName(class FString InPlayerName, int32_t InProductID, struct FScriptDelegate
Callback);
void InitWithPlayerID(struct FUniqueNetId InPlayerId, int32_t InProductID, struct
FProductInstanceID InInstanceId, struct FProductHashID InHashID, TArray<class
UProductAttribute_TA*> InAttributes, struct FScriptDelegate Callback);
void InitWithSaveObject();
void HandleRemovedOnlineProduct(class USaveData_TA* Data, class UOnlineProduct_TA*
OnlineProduct);
void HandleProfileDataChanged(class UPlayerVanitySave_TA* InData);
void InitWithProfile(class UProfile_TA* InProfile);
void EventEquipProduct(class UPlayerVanity_TA* PlayerVanity, struct FProductInstanceID
NewInstanceId, struct FProductInstanceID PrevInstanceId);
void EventOnLoadComplete(class UPlayerVanity_TA* PlayerVanity);
};


```

```

// Class TAGame.PlayerAvatar_TA
// 0x0008 (0x0198 - 0x01A0)
class UPlayerAvatar_TA : public UPlayerVanity_TA
{
public:
class UProductAsset_PlayerAvatar_TA*           Asset;          // 0x0198 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerAvatar_TA");
}

return uClassPointer;
};

void HandleLoaded(struct FAssetLoadResult Result);
void HandleUpdateTexture(class UTexture* AvatarTexture);
};


```

```
// Class TAGame.PlayerAvatarBorder_TA
// 0x0008 (0x0198 - 0x01A0)
class UPlayerAvatarBorder_TA : public UPlayerVanity_TA
{
public:
class UProductAsset_PlayerAvatarBorder_TA*      Asset;          // 0x0198
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerAvatarBorder_TA");
}

return uClassPointer;
};

void HandleLoaded(struct FAssetLoadResult Result);
class UClass* GetSaveType();
};

// Class TAGame.PlayerBanner_TA
// 0x0008 (0x0198 - 0x01A0)
class UPlayerBanner_TA : public UPlayerVanity_TA
{
public:
class UProductAsset_PlayerBanner_TA*      Asset;          // 0x0198 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerBanner_TA");
}

return uClassPointer;
};

void HandleLoaded(struct FAssetLoadResult Result);
class UClass* GetSaveType();
};

// Class TAGame.PremiumSkin_TA
// 0x0080 (0x0060 - 0x00E0)
class UPremiumSkin_TA : public UObject
{
```

```

public:
class UMaterialInstanceConstant*          MIC;           // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
uint8_t          Quality;        // 0x0068 (0x0001)
[0x0000000000000001] (CPF_Edit)
uint8_t          SkinType;       // 0x0069 (0x0001)
[0x0000000000000001] (CPF_Edit)
struct FMaterialParams          SkinParameters; // 0x0070 (0x0030)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UTexture2D*          Thumbnail;    // 0x00A0 (0x0008)
[0x0000000800000001] (CPF_Edit)
unsigned long      bTeamFinishDisabled : 1; // 0x00A8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long      bCustomFinishDisabled : 1; // 0x00A8 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
TArray<class UProductAssetAttribute_TA*> Attributes; // 0x00B0
(0x0010) [0x000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF>EditInline)
TArray<class UPremiumSkinSet_TA*> SkinSets; // 0x00C0
(0x0010) [0x0000000800420001] (CPF_Edit | CPF>EditConst | CPF_NeedCtorLink)
class UProductAsset_Skin_TA* AssetTemplate; // 0x00D0
(0x0008) [0x0000000804400008] (CPF_ExportObject | CPF_NeedCtorLink | CPF>EditInline)
class UProductTemplate_TA* ProductTemplate; // 0x00D8
(0x0008) [0x0000000804400008] (CPF_ExportObject | CPF_NeedCtorLink | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PremiumSkin_TA");
}

return uClassPointer;
};

// Class TAGame.PremiumSkinSet_TA
// 0x0018 (0x0060 - 0x0078)
class UPremiumSkinSet_TA : public UObject
{
public:
class UProductAsset_Body_TA* Body; // 0x0060 (0x0008)
[0x0000000800000001] (CPF_Edit)
TArray<struct FPremiumSkinSetItem> Skins; // 0x0068 (0x0010)
[0x0000000800400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PremiumSkinSet_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_TA
// 0x0098A0 (0x0060 - 0x00F8100)
class UProductAsset_TA : public UObject
{
public:
class UProductTemplate_TA*           Product;           // 0x0060 (0x0008)
[0x0000000804020001] (CPF>Edit | CPF>EditConst | CPF>EditInline)
class UTexture2D*                   Thumbnail;        // 0x0068 (0x0008)
[0x0002000800200003] (CPF>Edit | CPF>Const)
class UThumbnailRenderer_TA*        ThumbnailRenderer; // 0x0070
(0x0008) [0x0002000000000001] (CPF>Edit)
class UMaterialInterface*          ThumbnailMaterial; // 0x0078 (0x0008)
[0x0002000000000001] (CPF>Edit)
float                               ThumbnailKeyLightBrightnessScale; // 0x0080 (0x0004)
[0x0000000000000001] (CPF>Edit)
struct FVector                      ThumbnailCameraTranslationOffset; // 0x0084
(0x000C) [0x0000000000000003] (CPF>Edit | CPF>Const)
struct FRotator                     ThumbnailCameraRotationOffset; // 0x0090
(0x000C) [0x0000000000000003] (CPF>Edit | CPF>Const)
float                               ThumbnailCameraFOV;      // 0x009C (0x0004)
[0x0000000000000003] (CPF>Edit | CPF>Const)
struct FRotator                     ThumbnailProductRotation; // 0x00A0 (0x000C)
[0x0000000000000003] (CPF>Edit | CPF>Const)
struct FVector                      ThumbnailProductTranslationOffset; // 0x00AC
(0x000C) [0x0000000000000003] (CPF>Edit | CPF>Const)
struct FVector                      ThumbnailProductDrawScale3D; // 0x00B8
(0x000C) [0x0000000000000003] (CPF>Edit | CPF>Const)
float                               ParticleWarmupTime;    // 0x00C4 (0x0004)
[0x0000000000000003] (CPF>Edit | CPF>Const)
float                               AnimationWarmupTime; // 0x00C8 (0x0004)
[0x0000000000000003] (CPF>Edit | CPF>Const)
struct FVector                      ParticleFakeVelocity; // 0x00CC (0x000C)
[0x0000000000000003] (CPF>Edit | CPF>Const)
struct FVector                      SkeletalMeshAnimationOffset; // 0x00D8 (0x000C)
[0x0000000000000003] (CPF>Edit | CPF>Const)
unsigned long                      bForceRebuildThumbnail : 1; // 0x00E4 (0x0004)
[0x0000000800000003] [0x00000001] (CPF>Edit | CPF>Const)
unsigned long                      bPreviewDynamicThumbnail : 1; // 0x00E4 (0x0004)
[0x0000000800000003] [0x00000002] (CPF>Edit | CPF>Const)
unsigned long                      bServerSideOnlyWasSetInitially : 1; // 0x00E4 (0x0004)
[0x0000000800002000] [0x00000004] (CPF>Transient)
class UProductPaint_TA*            PaintToTest;       // 0x00E8 (0x0008)
[0x0000000800002003] (CPF>Edit | CPF>Const | CPF>Transient)
TArray<class UProductAssetAttribute_TA*> Attributes; // 0x00E8F0

```

(0x0010) [0x0000000004400001] (CPF\_Edit | CPF\_NeedCtorLink | CPF\_EditInline)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_TA");  
}  
  
return uClassPointer;  
};  
  
uint8_t eventCheckIfErrorOrWarning(class FString& OutString);  
void SetupDynamicThumbnail(class UPrimitiveComponent* PrimComp);  
class UTexture2D* RenderAssetThumbnail(class FString TextureName, class UPackage* InPackage);  
struct FName GetThumbnailAssetName();  
struct FName GetThumbnailAssetPackageName();  
class FString GetAssetPackageName();  
class FString GetProductPackageName();  
class UObject* GetAttribute(class UClass* AttributeClass);  
void AddFXActorToThumbnailScene(class AFXActor_X* FXActor, class UCarMeshComponent_TA* SKMToAttachTo, class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);  
bool CreateAndAttachThumbnailPSC(struct FFXAttachment Attachment, class AFXActor_X* FXActor, int32_t AttachmentIdx, class USkeletalMeshComponent* Owner, TArray<class UProductAttribute_TA*>& InAttributes);  
class UParticleSystem* GetParticleSystemThumbnailOverride(class AFXActor_X* FXActor, int32_t AttachmentIdx, struct FFXAttachment& Attachment);  
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);  
class UProduct_TA* GetProduct();  
};  
  
// Class TAGame.ProductAsset_Attachment_TA  
// 0x0018 (0x00F8100 - 0x01108)  
class UProductAsset_Attachment_TA : public UProductAsset_TA  
{  
public:  
TArray<struct FProductAttachment> Attachments; // 0x00F8100  
(0x0010) [0x00000000000480001] (CPF>Edit | CPF_Component | CPF_NeedCtorLink)  
struct FName CustomAnimName; // 0x01108 (0x0008)  
[0x0000000000000001] (CPF>Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Attachment_TA");
}

return uClassPointer;
};

void ModifyThumbnailMesh(class UMeshComponent* Component, struct FProductAttachment& Attachment, TArray<class UProductAttribute_TA*>& InAttributes);
class USkeletalMeshComponent* GetSceneSkeletalMesh(class UThumbnailScene_TA* Scene, int32_t AttachmentIdx);
class UStaticMeshComponent* GetSceneStaticMesh(class UThumbnailScene_TA* Scene, int32_t AttachmentIdx);
void PrepParticleSystem(class USkeletalMeshComponent* AttachSKC, class UParticleSystemComponent* PSC, struct FName SocketName, TArray<class UProductAttribute_TA*>& InAttributes);
void HandleAttachedParticleSystem(class UParticleSystemComponent* ParticleSystemAttachment, class USkeletalMeshComponent*& AttachSKC, class UStaticMeshComponent*& AttachSMC, struct FProductAttachment& Attachment, TArray<class UProductAttribute_TA*>& InAttributes);
void HandleAttachedSkeletalMesh(class USkeletalMeshComponent*& AttachSKC, class USkeletalMesh*& InSkeletalMesh, struct FProductAttachment& Attachment, TArray<class UProductAttribute_TA*>& InAttributes);
void HandleAttachedStaticMesh(class UStaticMeshComponent*& AttachSMC, class UStaticMesh*& InStaticMesh, struct FProductAttachment& Attachment, TArray<class UProductAttribute_TA*>& InAttributes);
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);
};

```

```

// Class TAGame.ProductAsset_Body_TA
// 0x0280 (0x00F8100 - 0x03780)
class UProductAsset_Body_TA : public UProductAsset_TA
{
public:
    class USkeletalMesh* Mesh; // 0x00F8100 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAnimTree* AnimTreeTemplate; // 0x01008 (0x0008)
    [0x0002000000000001] (CPF_Edit)
    TArray<class UAnimSet*> AnimSets; // 0x01108 (0x0010)
    [0x0002000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class UPhysicsAsset* PhysicsAsset; // 0x011820 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UHandlingPreset_TA* HandlingPreset; // 0x01208 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float HandlingPresetZOffset; // 0x012830 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FVehicleAxeSettings FrontAxe; // 0x012C34 (0x0028)
    [0x0000000000000001] (CPF_Edit)
    struct FVehicleAxeSettings BackAxe; // 0x0154C (0x0028)
    [0x0000000000000001] (CPF_Edit)
    TArray<struct FVehicleAxeSettings> AdditionalAxles; // 0x01808
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    float SteerScale; // 0x01908 (0x0004)
    [0x0000000000000001] (CPF_Edit)

```

```

struct FSimpleSpringSettings           ChassisSpring;          // 0x0194C (0x0050)
[0x0000000000000001] (CPF_Edit)
float                                ChassisRotationScale; // 0x01E4C (0x0004)
[0x0000000000220001] (CPF_Edit | CPF_EditConst)
float                                ChassisTranslationScale; // 0x01E8F0 (0x0004)
[0x0000000000220001] (CPF_Edit | CPF_EditConst)
float                                SuspensionTravelMax; // 0x01ECF4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                SuspensionTravelMin; // 0x01F08 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long                         bUseLegacySuspensionOffsets : 1; // 0x01F4C
(0x0004) [0x0000000000000000] [0x00000001]
unsigned long                         bOverrideBoostFlapsAngle : 1; // 0x01F4C (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long                         bUseForcedColors : 1; // 0x01F4C (0x0004)
[0x0000008000000001] [0x00000004] (CPF_Edit)
class UAkSoundCue*                  EngineSound;          // 0x01F8200 (0x0008)
[0x0000000020000000] CPF_Deprecated)
class UAkSoundCue*                  ExhaustSound;        // 0x02008 (0x0008)
[0x0000000020000000] CPF_Deprecated)
class UAkSoundCue*                  BlowoffSound;       // 0x02108 (0x0008)
[0x0000000020000000] CPF_Deprecated)
class UEngineAudioProfile_TA*      EngineAudioProfile; // 0x02108
(0x0008) [0x0000000020000000] CPF_Deprecated)
class UProductAsset_EngineAudio_TA* EngineAudioAsset;    // 0x021820
(0x0008) [0x0000000820000000] CPF_Deprecated)
TArray<class UProductAsset_EngineAudio_TA*> EngineAudioAssets; // 0x02208 (0x0010)
[0x0000000080040001] (CPF_Edit | CPF_NeedCtorLink)
int32_t                             EngineAudioProductID; // 0x02308 (0x0004)
[0x0000000020020000] (CPF>EditConst | CPF_Deprecated)
class UProductAssetReference_TA*   EngineAudioReference; // 0x023840
(0x0008) [0x00000000400001] (CPF>Edit | CPF>EditInline)
class AFXActor_X*                 FXActor;             // 0x02408 (0x0008)
[0x0002000000000001] (CPF>Edit)
int32_t                             SkinMaterialIndex; // 0x024850 (0x0004)
[0x0000000000000001] (CPF>Edit)
int32_t                             BrakelightMaterialIndex; // 0x0254C (0x0004)
[0x0000000000000001] (CPF>Edit)
int32_t                             ChassisMaterialIndex; // 0x02508 (0x0004)
[0x0000000000000001] (CPF>Edit)
int32_t                             BoostMaterialIndex; // 0x0254C (0x0004)
[0x0000000000000001] (CPF>Edit)
TArray<struct FName>              BoostEmitterSockets; // 0x025860
(0x0010) [0x00000000040001] (CPF>Edit | CPF_NeedCtorLink)
float                               BoostFlapsAngle; // 0x026870 (0x0004)
[0x0000000000000001] (CPF>Edit)
struct FLinearColor                ForcedTeamColors[0x2]; // 0x026C74
(0x0020) [0x0000000000000001] (CPF>Edit)
struct FLinearColor                ForcedCustomColor; // 0x028C94 (0x0010)
[0x0000000000000001] (CPF>Edit)
struct FVector                     SupersonicTrailOffset; // 0x029CA4 (0x000C)
[0x0000000000000001] (CPF>Edit)
TArray<class UProduct_TA*>        AdditionalLocks2; // 0x02A8B0
(0x0010) [0x0000000000040001] (CPF>Edit | CPF_NeedCtorLink)

```

```

class UProductEquipProfile_TA*           EquipProfile;           // 0x02B8C0
(0x0008) [0x000000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
TArray<struct FProductAttachment>       Attachments;          // 0x02C08
(0x0010) [0x000200000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
TArray<struct FParticleSysParam>        BoostParticleParameters; // 0x02D08
(0x0010) [0x000200000400001] (CPF_Edit | CPF_NeedCtorLink)
class UParticleSystem*                  SupersonicWheelTemplate; // 0x02E08
(0x0008) [0x0002000000000001] (CPF_Edit)
class AFXActor_X*                     DemolishFX;           // 0x02E8F0 (0x0008)
[0x0002000000000001] (CPF_Edit)
class AFXActor_X*                     SpawnFX;              // 0x02F08 (0x0008)
[0x0002000000000001] (CPF_Edit)
class AFXActor_X*                     ExitFX;               // 0x02F8300 (0x0008)
[0x0002000000000001] (CPF_Edit)
float                               WheelPreviewAnimTranslateScale; // 0x03008 (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FBoostAttachmentToggle>   BoostAttachmentsToModify; // 0x03108 (0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FMaterialParams                SkinParameters;         // 0x031820 (0x0030)
[0x000200000400001] (CPF_Edit | CPF_NeedCtorLink)
class UProductAssetReferenceWheel_TA*   ThumbnailWheelAsset;    //
0x034850 (0x0008) [0x0002000004000001] (CPF_Edit | CPF_EditInline)
class UProductAssetReferenceSkin_TA*    ThumbnailSkinAsset;    // 0x03508
(0x0008) [0x0002000004000001] (CPF_Edit | CPF_EditInline)
TArray<struct FSkeletalMeshLODDistanceInfo> LODDistanceInfo; // 0x035860 (0x0010) [0x000000000400000] (CPF_NeedCtorLink)
TArray<class UObject*>                 MutatorArchetypes;    // 0x036870
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Body_TA");
}

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
bool CanEquip(class UProduct_TA* InProduct);
bool HasBone(struct FName InBoneName);
void FixPhysicsBoxTranslation();
void ApplyBoostAttachmentDisablesToFXActor(class AFXActor_X* OutFXActor);
void ModifyAttachment(class AFXActor_X* OutFXActor, int32_t AttachmentIndex, struct
FBoostAttachmentToggle& EventsToUse);
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class
UProductAttribute_TA*>& InAttributes);
};

```

```

// Class TAGame.ProductAsset_Boost_TA
// 0x0018 (0x00F8100 - 0x01108)
class UProductAsset_Boost_TA : public UProductAsset_TA
{
public:
    class AFXActor_Boost_TA*           FXActor;          // 0x00F8100 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class AFXActor_Boost_TA*           PaintedFXActor;   // 0x01008 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float                             PreviewFrontWheelSpinSpeed; // 0x01108 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             PreviewBackWheelSpinSpeed; // 0x010C14 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Boost_TA");
        }

        return uClassPointer;
    };

    uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
    void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);
    class AFXActor_Boost_TA* GetBoostFXActorForMesh(class UCarMeshComponentBase_TA* CarMesh);
    class UParticleSystem* GetParticleSystemThumbnailOverride(class AFXActor_X* InFXActor, int32_t AttachmentIdx, struct FFXAttachment& Attachment);
    };
}

// Class TAGame.ProductAsset_Bot_TA
// 0x001C (0x00F8100 - 0x0114C)
class UProductAsset_Bot_TA : public UProductAsset_TA
{
public:
    TArray<class UProduct_TA*>           LoadoutProducts2;      // 0x00F8100
    (0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    int32_t                            BlueTeamColorID;       // 0x01108 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                            OrangeTeamColorID;    // 0x010C14 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                            CustomColorID;        // 0x01108 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    };
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Bot_TA");
}

return uClassPointer;
};

struct FLoadoutData GetLoadout(int32_t TeamIndex);
};

// Class TAGame.ProductAsset_Container_TA
// 0x0000 (0x00F8100 - 0x00F8100)
class UProductAsset_Container_TA : public UProductAsset_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Container_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_EngineAudio_TA
// 0x0040 (0x00F8100 - 0x013840)
class UProductAsset_EngineAudio_TA : public UProductAsset_TA
{
public:
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) EngineSound; // 0x00F8100 (0x0008)
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) ExhaustSound; // 0x01008 (0x0008)
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) IgnitionSound; // 0x01108 (0x0008)
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) BlowoffSound; // 0x01108 (0x0008)
class UAkSoundCue* [0x0000000000000001] (CPF_Edit) ThrottleEngageSound; // 0x011820
(0x0008) [0x0000000000000001] (CPF_Edit) ThrottleReleaseSound; // 0x01208
(0x0008) [0x0000000000000001] (CPF_Edit) GearChangeSound; // 0x012830
(0x0008) [0x0000000000000001] (CPF_Edit) Profile; // 0x01308 (0x0008)
[0x0000000000000001] (CPF_Edit)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_EngineAudio_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_GoalExplosion_TA
// 0x0028 (0x00F8100 - 0x01208)
class UProductAsset_GoalExplosion_TA : public UProductAsset_TA
{
public:
class AFXActor_X* GoalExplosionFX; // 0x00F8100 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X* PaintedGoalExplosionFX; // 0x01008 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<class UExplosionHitHandler_X*> ExplosionComponents; //
0x01108 (0x0010) [0x000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
unsigned long bUseOverrideGoalOrientations : 1; // 0x011820
(0x0004) [0x0000000000000001] [0x00000001] (CPF_Edit)
struct FIgnoredGoalRotations IgnoredRotations; // 0x011C24
(0x0004) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_GoalExplosion_TA");
}

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
void OverrideSpawnRotation(struct FRotator& SpawnRotation);
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);
class AFXActor_X* GetExplosionFXActorForPRI(class APRI_TA* PRI);
class AFXActor_X* GetExplosionFXActorForMesh(class UCarMeshComponentBase_TA* CarMesh);
};

```

```

// Class TAGame.ProductAsset_Pack_TA
// 0x0000 (0x00F8100 - 0x00F8100)
class UProductAsset_Pack_TA : public UProductAsset_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Pack_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_PaintFinish_TA
// 0x0060 (0x00F8100 - 0x015860)
class UProductAsset_PaintFinish_TA : public UProductAsset_TA
{
public:
class ULightCurveType_TA*           LightCurveType;           // 0x00F8100
(0x0008) [0x0000000000000001] (CPF_Edit)
class UTexture*                   DetailNormal;           // 0x01008 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                           SpecularTint;           // 0x01108 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           RimLightTint;           // 0x010C14 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           DiffuseDetailNormalStrength; // 0x01108 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           SpecularDetailNormalStrength; // 0x0114C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           SpecularStrength;        // 0x011820 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           EnvironmentStrength; // 0x011C24 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           SparkleStrength;        // 0x01208 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                           PearlescentStrength; // 0x0124C (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FTextureParameterValue>   TextureParameterValues; // 
0x012830 (0x0010) [0x00000000420003] (CPF_Edit | CPF_Const | CPF_EditConst | 
CPF_NeedCtorLink)
TArray<struct FScalarParameterValue>   ScalarParameterValues; // 
0x013840 (0x0010) [0x00000000420003] (CPF_Edit | CPF_Const | CPF_EditConst | 
CPF_NeedCtorLink)
TArray<struct FVectorParameterValue>   VectorParameterValues; // 

```

0x014850 (0x0010) [0x000000000420003] (CPF\_Edit | CPF\_Const | CPF\_EditConst | CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
    static UClass* uClassPointer = nullptr;  
  
    if (!uClassPointer)  
    {  
        uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_PaintFinish_TA");  
    }  
  
    return uClassPointer;  
};  
  
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class  
UProductAttribute_TA*>& InAttributes);  
static void SetPaintFinishParametersOnMaterial(class UMaterialInstance* MatInst, class  
UProductAsset_PaintFinish_TA* Finish, class FString Prefix);  
};  
  
// Class TAGame.ProductAsset_PlayerVanity_TA  
// 0x0030 (0x00F8100 - 0x012830)  
class UProductAsset_PlayerVanity_TA : public UProductAsset_TA  
{  
public:  
    class USwfMovie* [0x0000000000000001] (CPF_Edit) MovieReference; // 0x00F8100 (0x0008)  
    class UTexture2D* [0x0000000000000001] (CPF_Edit) ToPlayer; // 0x01008 (0x0008)  
    class UTexture2D* [0x0000000000000001] (CPF_Edit) PaintableLayer; // 0x01108 (0x0008)  
    class UTexture2D* [0x0000000000000001] (CPF_Edit) TintableLayer; // 0x01108 (0x0008)  
    struct FLinearColor [0x0000000000000003] (CPF_Edit | CPF_Const) DefaultPaintColor; // 0x011820 (0x0010)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_PlayerVanity_TA");  
        }  
  
        return uClassPointer;  
    };  
};  
  
// Class TAGame.ProductAsset_PlayerAvatar_TA
```

```
// 0x0000 (0x012830 - 0x012830)
class UProductAsset_PlayerAvatar_TA : public UProductAsset_PlayerVanity_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_PlayerAvatar_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_PlayerAvatarBorder_TA
// 0x0000 (0x012830 - 0x012830)
class UProductAsset_PlayerAvatarBorder_TA : public UProductAsset_PlayerVanity_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_PlayerAvatarBorder_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_PlayerBanner_TA
// 0x0000 (0x012830 - 0x012830)
class UProductAsset_PlayerBanner_TA : public UProductAsset_PlayerVanity_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_PlayerBanner_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_Skin_TA
// 0x0110 (0x00F8100 - 0x02108)
class UProductAsset_Skin_TA : public UProductAsset_TA
{
public:
    class UMaterialInterface*           Skin;           // 0x00F8100 (0x0008)
    [0x0000000000000001] (CPF_Edit) 
    TArray<struct FSkinBodySettings>   BodySettings;   // 0x01008
    (0x0010) [0x0000000000040001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<struct FBodyChassisOverride> BodyChassisOverrides; // 0x01108
    (0x0010) [0x000000000040001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<struct FProductAttachment>   Attachments;   // 0x01208
    (0x0010) [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
    uint8_t                           SkinType;       // 0x01308 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    struct FLinearColor               ThumbnailTeamColor; // 0x0134C (0x0010)
    [0x0000000000000001] (CPF_Edit)
    struct FLinearColor               ThumbnailCustomColor; // 0x0144C (0x0010)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                     bTeamFinishDisabled : 1; // 0x0154C (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long                     bCustomFinishDisabled : 1; // 0x0154C (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    struct FParameterInformation     TeamFinishToApply; // 0x015860
    (0x0038) [0x0000000000040001] (CPF_Edit | CPF_NeedCtorLink)
    struct FParameterInformation     CustomFinishToApply; // 0x01908
    (0x0038) [0x0000000000040001] (CPF_Edit | CPF_NeedCtorLink)
    struct FLinearColor              ForcedTeamColors[0x2]; // 0x01C8D0
    (0x0020) [0x0000000000000001] (CPF_Edit)
    struct FLinearColor              ForcedCustomColor[0x2]; // 0x01E8F0
    (0x0020) [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Skin_TA");
        }

        return uClassPointer;
    };

    uint8_t eventCheckIfErrorOrWarning(class FString& OutString);

```

```

bool IsAnimatedSkinType();
bool HasForcedCustomColor(int32_t TeamIndex);
bool HasForcedTeamColor(int32_t TeamIndex);
bool HasForcedCustomFinish();
bool HasForcedTeamFinish();
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);
int32_t GetThumbnailBodyID(class UProfile_TA* Profile, int32_t EquippedBodyID);
struct FSkinBodySettings GetSkinBodySettings(class UProductAsset_Body_TA* ForBody);
struct FMaterialParams GetSkinParameters(class UProductAsset_Body_TA* ForBody);
void AttemptApplyChassisOverride(class UCarMeshComponentBase_TA* CarMesh, class UProductAsset_Body_TA*& ForBody);
};

// Class TAGame.ProductAsset_SupersonicTrail_TA
// 0x0028 (0x00F8100 - 0x01208)
class UProductAsset_SupersonicTrail_TA : public UProductAsset_TA
{
public:
    class AFXActor_TA* [0x0000000000000002] (CPF_Const) FXActor; // 0x00F8100 (0x0008)
    class AFXActor_TA* [0x0000000000000001] (CPF_Edit) LeftFXActor; // 0x01008 (0x0008)
    class AFXActor_TA* [0x0000000000000001] (CPF_Edit) RightFXActor; // 0x01108 (0x0008)
    class UAkSoundCue* [0x0000000000000001] (CPF_Edit) AkEnterSound; // 0x01108 (0x0008)
    class UAkSoundCue* [0x0000000000000001] (CPF_Edit) AkLoopSound; // 0x011820 (0x0008)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_SupersonicTrail_TA");
        }

        return uClassPointer;
    }

    void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);
};

// Class TAGame.ProductAsset_Wheel_TA
// 0x00B0 (0x00F8100 - 0x01A8B0)
class UProductAsset_Wheel_TA : public UProductAsset_TA
{
public:
    float [0x0000000000000001] (CPF_Edit) MeshRadius; // 0x00F8100 (0x0004)
}

```

```

float MeshWidth; // 0x00FC104 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UWheelAssetGenerator_TA* Generator; // 0x01008
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UWheelAssetGenerator_TA* PrevGenerator; // 0x01108
(0x0008) [0x00000080408000A] (CPF_Const | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAkSoundCue* CustomWheelDriveSound; // 0x01108
(0x0008) [0x0000000000000001] (CPF_Edit)
class UAkSoundCue* CustomWheelImpactSound; // 0x011820
(0x0008) [0x0000000000000001] (CPF_Edit)
uint8_t MirrorMode; // 0x01208 (0x0001)
[0x0000000000000002] (CPF_Const)
class UStaticMesh* StaticMesh; // 0x012830 (0x0008)
[0x0000000000000002] (CPF_Const)
class UStaticMesh* RearStaticMesh; // 0x01308 (0x0008)
[0x0000000000000002] (CPF_Const)
class USkeletalMesh* SkeletalMesh; // 0x013840 (0x0008)
[0x0000000000000002] (CPF_Const)
class USkeletalMesh* RearSkeletalMesh; // 0x01408 (0x0008)
[0x0000000000000002] (CPF_Const)
class UAnimTree* AnimTree; // 0x014850 (0x0008)
[0x0000000000000002] (CPF_Const)
class UAnimTree* LeftAnimTree; // 0x01508 (0x0008)
[0x0000000000000002] (CPF_Const)
TArray<class UAttachmentBehavior_TA*> Behaviors; // 0x015860
(0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FWheelAttachment Attachment; // 0x016870 (0x0030)
[0x0000000000480002] (CPF_Const | CPF_Component | CPF_NeedCtorLink)
TArray<struct FWheelAttachment> Attachments; // 0x0198A0
(0x0010) [0x0000000000480002] (CPF_Const | CPF_Component | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Wheel_TA");
}

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
void ApplyPaintToWheelAttachment(class UPrimitiveComponent* PrimComp, struct FWheelAttachment InAttachment, TArray<class UProductAttribute_TA*>& InAttributes);
void eventModifyThumbnailScene(class UThumbnailScene_TA* Scene, TArray<class UProductAttribute_TA*>& InAttributes);
};

// Class TAGame.ProductAssetAttribute_TA

```

```

// 0x0000 (0x0060 - 0x0060)
class UProductAssetAttribute_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetAttribute_TA");
}

return uClassPointer;
};

class UProductAttribute_TA* eventGenerateProductAttribute(class UObject* InParent);
bool eventRequiresGeneratedProductAttribute();
uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
};

// Class TAGame.AssetAttribute_TeamEdition_TA
// 0x0024 (0x0060 - 0x0084)
class UAssetAttribute_TeamEdition_TA : public UProductAssetAttribute_TA
{
public:
TArray<class UProductOverride_TA*>           Template;           // 0x0060
(0x0010) [0x0000000804400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInline)
TArray<struct FTeamEditionOverrides>           TeamEditions;        // 0x0070
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long          bAddAllAvailableTeams : 1;           // 0x0080 (0x0004)
[0x0000000800000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AssetAttribute_TeamEdition_TA");
}

return uClassPointer;
};

void eventApplyPaintOverrides(int32_t TeamID, class UProductPaint_TA* Paint, class UObject* Target);
void eventApplyToObject(int32_t TeamID, class UObject* Target);
int32_t GetIndexFromTeamID(int32_t TeamID);
};

```

```

// Class TAGame.AssetAttribute_TeamOverride_TA
// 0x0028 (0x0060 - 0x0088)
class UAssetAttribute_TeamOverride_TA : public UProductAssetAttribute_TA
{
public:
    uint8_t Team; // 0x0060 (0x0001)
    [0x0000000000000001] (CPF_Edit) TArray<class UMaterialInterface*> Materials; // 0x0068 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    struct FName AssetName; // 0x0078 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UTexture* Thumbnail; // 0x0080 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AssetAttribute_TeamOverride_TA");
        }

        return uClassPointer;
    };

    class UProductAttribute_TA* eventGenerateProductAttribute(class UObject* InParent);
    bool eventRequiresGeneratedProductAttribute();
};

// Class TAGame.AssetAttribute_TeamParameterOverride_TA
// 0x0038 (0x0060 - 0x0098)
class UAssetAttribute_TeamParameterOverride_TA : public UProductAssetAttribute_TA
{
public:
    uint8_t Team; // 0x0060 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    struct FMaterialParams ParamOverrides; // 0x0068 (0x0030)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.AssetAttribute_TeamParameterOverride_TA");
        }

        return uClassPointer;
    };
}

```

```

class UProductAttribute_TA* eventGenerateProductAttribute(class UObject* InParent);
bool eventRequiresGeneratedProductAttribute();
};

// Class TAGame.ProductAttribute_PaintSettings_TA
// 0x00B8 (0x0060 - 0x0118)
class UProductAttribute_PaintSettings_TA : public UProductAssetAttribute_TA
{
public:
    struct FPaintMaterialGroup           MaterialGroups[0x2];           // 0x0060 (0x0020)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    unsigned long                      bPaintParticles : 1;          // 0x0080 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long                      bPaintBody : 1;              // 0x0080 (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    unsigned long                      bGammaCorrect : 1;           // 0x0080 (0x0004)
    [0x0000000000000001] [0x00000004] (CPF_Edit)
    struct FName                        PaintParameterName;           // 0x0084 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    uint8_t                            PaintType;                  // 0x008C (0x0001)
    [0x0000000000000001] (CPF_Edit)
    TArray<struct FPaintAttributeParameter> PaintAdditionalParameters; // 0x0090
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    float                             PaintEmissiveMultiplier;        // 0x00A0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    TArray<class UMaterialInterface*> PaintableMaterials;           // 0x00A8
    (0x0010) [0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
    TArray<class UMaterialInterface*> PaintableMaterialsMetallic; // 0x00B8
    (0x0010) [0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
    TArray<class UProductPaint_TA*>   IncludePaints;                // 0x00C8
    (0x0010) [0x0000000800402003] (CPF_Edit | CPF_Const | CPF_Transient | CPF_NeedCtorLink)
    TArray<class UProductPaint_TA*>   ExcludePaints;                // 0x00D8
    (0x0010) [0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
    TArray<class UProductPaint_TA*>   UnsupportedPaints;             // 0x00E8
    (0x0010) [0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
    TArray<class UProductPaint_TA*>   ShopExclusivePaints;           // 0x00F8
    (0x0010) [0x0000000800400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
    TArray<struct FPaintWithOverride> PaintsToOverride;               // 0x0108
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_PaintSettings_TA");
        }

        return uClassPointer;
    };
}

```

```

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
class UProductPaint_TA* GetPaintFromProductOverride(class UProductOverride_TA*
ProductOverrideToFind);
};

// Class TAGame.ProductAssetLoader_TA
// 0x0018 (0x0060 - 0x0078)
class UProductAssetLoader_TA : public UObject
{
public:
unsigned long          bDebug : 1;           // 0x0060 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bLoadingAssetAsync : 1;    // 0x0060 (0x0004)
[0x0000000000002002] [0x00000002] (CPF_Const | CPF_Transient)
unsigned long          bAssetLoadedThisFrame : 1;   // 0x0060 (0x0004)
[0x0000000000002002] [0x00000004] (CPF_Const | CPF_Transient)
TArray<struct FAssetLoadRequest>      LoadAssetQueue;        // 0x0068
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetLoader_TA");
}

return uClassPointer;
};

void Tick(float DeltaTime);
void eventSetTickEnabled(unsigned long bTick);
void FlushAsyncLoading();
void CancelLoadAsset(int32_t ProductID, class UObject* Listener);
void CancelLoadAssets(class UObject* Listener);
void LoadAssetByNameAsync(struct FName AssetName, struct FScriptDelegate OnLoaded);
void LoadThumbnailAssetAsync(int32_t ProductID, struct FScriptDelegate OnLoaded);
void LoadAssetAsync(int32_t ProductID, struct FScriptDelegate OnLoaded);
class UProductAsset_TA* LoadAssetByName(struct FName AssetName);
class UProductAsset_TA* LoadAsset(int32_t ProductID);
static class UProductAssetLoader_TA* GetInstance();
};

// Class TAGame.ProductAssetReferenceBase_TA
// 0x000C (0x0060 - 0x006C)
class UProductAssetReferenceBase_TA : public UObject
{
public:
class UProductAsset_TA*          ProductAsset;        // 0x0060 (0x0008)
[0x0000000800000000]
int32_t              ProductID;           // 0x0068 (0x0004)
[0x00000000000020001] (CPF_Edit | CPF_EditConst)

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetReferenceBase_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAssetReference_TA
// 0x000C (0x006C - 0x0078)
class UProductAssetReference_TA : public UProductAssetReferenceBase_TA
{
public:
class UProductAsset_TA*           Asset;           // 0x0070 (0x0008)
[0x0000000800000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetReference_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAssetReferenceBody_TA
// 0x000C (0x006C - 0x0078)
class UProductAssetReferenceBody_TA : public UProductAssetReferenceBase_TA
{
public:
class UProductAsset_Body_TA*      BodyAsset;       // 0x0070 (0x0008)
[0x0000000800000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.ProductAssetReferenceBody_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAssetReferencePaintFinish_TA
// 0x000C (0x006C - 0x0078)
class UProductAssetReferencePaintFinish_TA : public UProductAssetReferenceBase_TA
{
public:
class UProductAsset_PaintFinish_TA*      PaintFinishAsset;           // 0x0070
(0x0008) [0x0000000800000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetReferencePaintFinish_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAssetReferenceSkin_TA
// 0x000C (0x006C - 0x0078)
class UProductAssetReferenceSkin_TA : public UProductAssetReferenceBase_TA
{
public:
class UProductAsset_Skin_TA*      SkinAsset;                // 0x0070 (0x0008)
[0x0000000800000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetReferenceSkin_TA");
}

return uClassPointer;
};

};

```

```

// Class TAGame.ProductAssetReferenceWheel_TA
// 0x000C (0x006C - 0x0078)
class UProductAssetReferenceWheel_TA : public UProductAssetReferenceBase_TA
{
public:
class UProductAsset_Wheel_TA*           WheelAsset;          // 0x0070
(0x0008) [0x0000000800000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetReferenceWheel_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_TA
// 0x0020 (0x0060 - 0x0080)
class UProductAttribute_TA : public UObject
{
public:
struct FName             Typename;          // 0x0060 (0x0008)
[0x0000000000000002] (CPF_Const)
unsigned long            bReplicated : 1;    // 0x0068 (0x0004)
[0x0000000000000002] [0x00000001] (CPF_Const)
unsigned long            bRequiresDynamicThumbnail : 1; // 0x0068
(0x0004) [0x0000000000000002] [0x00000002] (CPF_Const)
unsigned long            bDummyAttribute : 1;    // 0x0068 (0x0004)
[0x0000000000000002] [0x00000004] (CPF_Const)
class FString            Label;             // 0x0070 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_TA");
}

return uClassPointer;
};

static class UProductAttribute_SpecialEdition_TA*
__ProductAttribute_TA__ModifyLoadout_0x1(struct FClientLoadoutOnlineProduct P);

```

```

bool HasLoadedAllAdditionalReferences(class ULoadingProduct_TA* ProductData);
void LoadAdditionalAssets(struct FScriptDelegate Callback);
bool ShouldLoadAdditionalAssets();
class FString eventGetSortLabel();
class FString GetTypeHashID();
class FString eventGetHashID();
class FString GetOnlineProductAttributeValue();
struct FOnlineProductAttribute InstanceOnlineProductAttribute();
static TArray<int32_t> ModifyLoadout(TArray<int32_t> OnlineProductIDs, TArray<int32_t>& LoadoutProductIDs, struct FClientLoadoutOnlineData& OnlineLoadout);
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
static int32_t SortAttributes(class UProductAttribute_TA* A, class UProductAttribute_TA* B);
static void ApplyAttributeOverrides(class UProductAsset_TA* Asset, class UObject* Target, TArray<class UProductAttribute_TA*>& Attributes, TArray<class UClass*>& AttributesToIgnore);
static void ApplyAttributes(class UProductAsset_TA* Asset, class UObject* Target, TArray<class UProductAttribute_TA*>& Attributes, TArray<class UClass*>& AttributesToIgnore);
void OnLoaded();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& OtherAttributes);
bool eventInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& OtherAttributes);
static class UProductAttribute_TA* Get(class UClass* AttributeClass, TArray<class UProductAttribute_TA*> Attributes);
static TArray<class UProductSlot_TA*> eventGetClientAuthoritativeSlots();
uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
};

// Class TAGame.ProductAttribute_AnimatedSkinLabel_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_AnimatedSkinLabel_TA : public UProductAttribute_TA
{
public:
    class FString                     AnimatedLabel;           // 0x0080 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_AnimatedSkinLabel_TA");
        }

        return uClassPointer;
    };
};

// Class TAGame.ProductAttribute_Blueprint_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_Blueprint_TA : public UProductAttribute_TA
{

```

```
public:
int32_t ProductID; // 0x0080 (0x0004)
[0x0000004000000000]
int32_t CachedBlueprintSeriesID; // 0x0084 (0x0004)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Blueprint_TA");
}

return uClassPointer;
};

void __ProductAttribute_Blueprint_TA__OnInit_0x1(class UGFxData_ContainerDrops_TA* _);
void UpdateBlueprintSeriesID();
class FString GetSortLabel();
class FString GetTypeHashID();
class FString eventGetHashID();
class FString GetOnlineProductAttributeValue();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& OtherAttributes);
};

// Class TAGame.ProductAttribute_BodyCompatibility_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_BodyCompatibility_TA : public UProductAttribute_TA
{
public:
TArray<class UProduct_TA*> CompatibleBodies; // 0x0080
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_BodyCompatibility_TA");
}

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
};

// Class TAGame.ProductAttribute_Certified_TA
```

```

// 0x0030 (0x0080 - 0x00B0)
class UProductAttribute_Certified_TA : public UProductAttribute_TA
{
public:
    struct FName           ValueKeyName;          // 0x0080 (0x0008)
    [0x0000000000000002] (CPF_Const)
    class UCertifiedStat_TA*   CertifiedStat;        // 0x0088 (0x0008)
    [0x0000004000002000] (CPF_Transient)
    int32_t                StatId;              // 0x0090 (0x0004)
    [0x0000004000000000]
    int32_t                StatValue;            // 0x0094 (0x0004)
    [0x0000004000000000]
    struct FScriptDelegate      _EventRankedUp__Delegate; // 0x0098
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Certified_TA");
        }

        return uClassPointer;
    };

    class FString GetSortLabel();
    class FString eventGetHashID();
    class FString GetTypeHashID();
    class FString GetDescription();
    class FString GetRankLabel();
    int32_t GetRank();
    void AddStatValue(int32_t InValue);
    class FString GetOnlineProductAttributeValue();
    void OnLoaded();
    bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
    static class UProductAttribute_Certified_TA* CreateInstance(struct FName StatName);
    void EventRankedUp(class UProductAttribute_Certified_TA* CertifiedAttribute);
};

// Class TAGame.ProductAttribute_CompatibleProducts_TA
// 0x0018 (0x0080 - 0x0098)
class UProductAttribute_CompatibleProducts_TA : public UProductAttribute_TA
{
public:
    struct FPointer           VfTable_IICanEquip_TA; // 0x0080 (0x0008)
    [0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
    TArray<class UProduct_TA*>   CompatibleProducts; // 0x0088
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_CompatibleProducts_TA");
}

return uClassPointer;
};

bool CanEquip(TArray<int32_t>& EquippedProducts);
};

// Class TAGame.ProductAttribute_Container_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_Container_TA : public UProductAttribute_TA
{
public:
unsigned long          bUnlocked : 1;           // 0x0080 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bSkipRoulette : 1;        // 0x0080 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long          bGiftbox : 1;            // 0x0080 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Container_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_CrossEntitled_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_CrossEntitled_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_CrossEntitled_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_Currency_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_Currency_TA : public UProductAttribute_TA
{
public:
int32_t           CurrencyID;          // 0x0080 (0x0004)
[0x0000004000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Currency_TA");
}

return uClassPointer;
};

class FString GetSortLabel();
class FString GetTypeHashID();
class FString eventGetHashID();
class FString GetOnlineProductAttributeValue();
bool OnlInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& OtherAttributes);
};

// Class TAGame.ProductAttribute_Key_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_Key_TA : public UProductAttribute_TA
{
public:
unsigned long      bMagicKey : 1;        // 0x0080 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Key_TA");
}

```

```

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_Licensed_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_Licensed_TA : public UProductAttribute_TA
{
public:
class UProduct_TA* ReplacementProduct; // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Licensed_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_LicenseGroup_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_LicenseGroup_TA : public UProductAttribute_TA
{
public:
uint8_t Group; // 0x0080 (0x0001)
[0x0000000000000001] (CPF_Edit)
unsigned long bRequireSameLicenseGroupOnBody : 1; // 0x0084
(0x0004) [0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_LicenseGroup_TA");
}

return uClassPointer;
};

};

```

```

// Class TAGame.ProductAttribute_Painted_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_Painted_TA : public UProductAttribute_TA
{
public:
int32_t PaintID; // 0x0080 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Painted_TA");
}

return uClassPointer;
};

static TArray<class UProductOverride_TA*>
__ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x2(struct FPaintWithOverride O);
static bool __ProductAttribute_Painted_TA__ApplyToSetParameter_0x1(struct FPaintAttributeParameter PaintParam);
static bool CanApplyOverrideToSkin(class UProductOverride_MaterialParameter_TA* Override,
class UCarMeshComponent_TA* CarMeshComponent);
class FString GetSortLabel();
class FString eventGetHashID();
class FString GetTypeHashID();
void ApplyToPlayerVanity(class UPlayerVanity_TA* Vanity, class UProductPaint_TA* Paint);
static void ApplyToFXActor(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, class AFXActor_TA* FXActor);
static struct FLinearColor GetPaintColor(class UProductAttribute_PaintSettings_TA*
PaintSettings, class UProductPaint_TA* Paint, uint8_t PaintVariant, unsigned long bGammaCorrect);
static void ApplyToSetParameter(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, class UISetParameter* Target);
static void OverrideMeshMaterial(class UProductAttribute_PaintSettings_TA* PaintSettings,
class UProductPaint_TA* Paint, class UMeshComponent* Mesh, int32_t InMaterialIndex, class UMaterialInterface* InMaterial);
static void ApplyToBody(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, class UCarMeshComponentBase_TA* Mesh);
static void ApplyToSkin(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, class UCarMeshComponentBase_TA* Mesh);
static void ApplyToMesh(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, class UMeshComponent* Mesh, TArray<class UMaterialInterface*> InPaintMats);
static void ApplyPaintOverridesToObject(class UProductAttribute_PaintSettings_TA*
PaintSettings, class UProductPaint_TA* Paint, class UObject* Target);
static void ApplyPaintToObject(class UProductAttribute_PaintSettings_TA* PaintSettings, class UProductPaint_TA* Paint, class UObject* Target, class UProductAsset_TA* Asset);
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);

```

```

class FString GetOnlineProductAttributeValue();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
static class UProductAttribute_Painted_TA* CreateInstance(int32_t InPaintID);
};

// Class TAGame.ProductAttribute_PaintSettingsUpload_TA
// 0x0020 (0x0080 - 0x00A0)
class UProductAttribute_PaintSettingsUpload_TA : public UProductAttribute_TA
{
public:
TArray<int32_t> ExcludePaints; // 0x0080 (0x0010)
[0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
TArray<int32_t> ShopExclusivePaints; // 0x0090 (0x0010)
[0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_PaintSettingsUpload_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_PlatformExclusive_TA
// 0x0018 (0x0080 - 0x0098)
class UProductAttribute_PlatformExclusive_TA : public UProductAttribute_TA
{
public:
TArray<uint8_t> InclusiveConsoles; // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UProduct_TA* ReplacementProduct; // 0x0090 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_PlatformExclusive_TA");
}

return uClassPointer;
};

};

```

```
// Class TAGame.ProductAttribute_PostMatchCelebration_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_PostMatchCelebration_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_PostMatchCelebration_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_PreviewOnly_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_PreviewOnly_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_PreviewOnly_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_Quality_TA
// 0x0001 (0x0080 - 0x0081)
class UProductAttribute_Quality_TA : public UProductAttribute_TA
{
public:
uint8_t           Quality;           // 0x0080 (0x0001)
[0x0000004000000000]

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Quality_TA");
}

return uClassPointer;
};

static class FString ProductQualityToString(uint8_t InQuality);
class FString eventGetHashID();
class FString GetTypeHashID();
class FString GetOnlineProductAttributeValue();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
};

// Class TAGame.ProductAttribute_QualityProductDBOverride_TA
// 0x0001 (0x0080 - 0x0081)
class UProductAttribute_QualityProductDBOverride_TA : public UProductAttribute_TA
{
public:
uint8_t           QualityOverride;          // 0x0080 (0x0001)
[0x00000000000020001] (CPF_Edit | CPF_EditConst)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_QualityProductDBOverride_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_Reactive_TA
// 0x0014 (0x0080 - 0x0094)
class UProductAttribute_Reactive_TA : public UProductAttribute_TA
{
public:
class FString           ReactiveLabelKey;    // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long           bHasReactivePreview : 1; // 0x0090 (0x0004)
[0x00000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Reactive_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.ProductAttribute_RequiresOwnedProduct_TA  
// 0x0008 (0x0080 - 0x0088)  
class UProductAttribute_RequiresOwnedProduct_TA : public UProductAttribute_TA  
{  
public:  
    class UProduct_TA*           Product;           // 0x0080 (0x0008)  
    [0x0000000000000001] (CPF_Edit)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class  
TAGame.ProductAttribute_RequiresOwnedProduct_TA");  
        }  
  
        return uClassPointer;  
    };  
  
};  
  
// Class TAGame.ProductAttribute_Schematic_TA  
// 0x0000 (0x0080 - 0x0080)  
class UProductAttribute_Schematic_TA : public UProductAttribute_TA  
{  
public:  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Schematic_TA");  
        }  
  
        return uClassPointer;  
    };
```

```
};

};

// Class TAGame.ProductAttribute_SortingLabel_TA
// 0x0044 (0x0080 - 0x00C4)
class UProductAttribute_SortingLabel_TA : public UProductAttribute_TA
{
public:
    class FString           LocalizationKey;          // 0x0080 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString           ReplacementKey;         // 0x0090 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString           ReplacementValue;       // 0x00A0 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString           SortingLabel;          // 0x00B0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t                MaxReplacementSize;     // 0x00C0 (0x0004)
    [0x0000000000000002] (CPF_Const)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_SortingLabel_TA");
        }

        return uClassPointer;
    };

    void ReplaceKeyWithValue(class FString Value, class FString& InLabel);
    void UpdateSortingLabel(class FString& InLabel);
};

// Class TAGame.ProductAttribute_SpecialEdition_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_SpecialEdition_TA : public UProductAttribute_TA
{
public:
    int32_t                EditionID;             // 0x0080 (0x0004)
    [0x0000004000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_SpecialEdition_TA");
        }

    }
}
```

```

return uClassPointer;
};

bool __ProductAttribute_SpecialEdition_TA__GetOverrideProductID_0x1(struct
FSpecialEditionConfig E);
class FString GetSortLabel();
class FString eventGetHashID();
class FString GetTypeHashID();
class FString GetOnlineProductAttributeValue();
int32_t GetOverrideProductID(int32_t ProductID);
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
static class UProductAttribute_SpecialEdition_TA* CreateInstance(int32_t InEditionID);
};

// Class TAGame.ProductAttribute_SpecialEditionSettings_TA
// 0x0020 (0x0080 - 0x00A0)
class UProductAttribute_SpecialEditionSettings_TA : public UProductAttribute_TA
{
public:
TArray<struct FSpecialEditionConfig> Editions; // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UProductSpecialEdition_TA*> SupportedEditions; // 0x0090
(0x0010) [0x0000000800402003] (CPF>Edit | CPF_Const | CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_SpecialEditionSettings_TA");
}

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
};

// Class TAGame.ProductAttribute_TeamEdition_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_TeamEdition_TA : public UProductAttribute_TA
{
public:
int32_t Id; // 0x0080 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_TeamEdition_TA");
}

return uClassPointer;
};

bool HasLoadedAllAdditionalReferences(class ULoadingProduct_TA* ProductData);
void LoadAdditionalAssets(struct FScriptDelegate Callback);
bool ShouldLoadAdditionalAssets();
class FString GetSortLabel();
class FString GetTypeHashID();
class FString eventGetHashID();
void ApplyPaintOverrides(class UProductPaint_TA* Paint, class UProductAsset_TA* Asset, class
	UObject* Target);
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
class FString GetOnlineProductAttributeValue();
static class UProductAttribute_TeamEdition_TA* CreateInstance(int32_t TeamID);
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>&
OtherAttributes);
};

// Class TAGame.ProductAttribute_TeamEditionUpload_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_TeamEditionUpload_TA : public UProductAttribute_TA
{
public:
TArray<int32_t> SupportedTeamEditions; // 0x0080 (0x0010)
[0x0000000800420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_TeamEditionUpload_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_TeamForcedProduct_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_TeamForcedProduct_TA : public UProductAttribute_TA
{
public:
class UProductAssetReference_TA* Teams[0x2]; // 0x0080
(0x0010) [0x0000000004000001] (CPF_Edit | CPF_EditInline)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_TeamForcedProduct_TA");
}

return uClassPointer;
};

int32_t ValidateProduct(int32_t TeamIndex, int32_t ProductID);
bool eventCanEquip(int32_t ProductID);
};

// Class TAGame.ProductAttribute_TeamOverride_TA
// 0x000C (0x0080 - 0x008C)
class UProductAttribute_TeamOverride_TA : public UProductAttribute_TA
{
public:
uint8_t Team; // 0x0080 (0x0001)
[0x0000000000000001] (CPF_Edit)
struct FName AssetName; // 0x0084 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_TeamOverride_TA");
}

return uClassPointer;
};

static void ApplyToMesh(class UMeshComponent* Mesh, TArray<class UMaterialInterface*>& InMaterials);
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
};

// Class TAGame.ProductAttribute_TeamParameterOverride_TA
// 0x0001 (0x0080 - 0x0081)
class UProductAttribute_TeamParameterOverride_TA : public UProductAttribute_TA
{
public:
uint8_t Team; // 0x0080 (0x0001)
[0x0000000000000001] (CPF_Edit)

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_TeamParameterOverride_TA");
}

return uClassPointer;
};

static void ApplyToMesh(class UMeshComponent* TargetMesh, struct FMaterialParams&
InParams);
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
static bool HasParametersSpecified(struct FMaterialParams& InParams);
};

// Class TAGame.ProductAttribute_Tiered_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_Tiered_TA : public UProductAttribute_TA
{
public:
int32_t TierIndex; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Tiered_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_TitleID_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_TitleID_TA : public UProductAttribute_TA
{
public:
struct FName TitleId; // 0x0080 (0x0008)
[0x0000040000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_TitleID_TA");
}

return uClassPointer;
};

class FString eventGetHashID();
class FString GetTypeHashID();
class FString GetOnlineProductAttributeValue();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& OtherAttributes);
};

// Class TAGame.ProductAttribute_UnlockForcedProducts_TA
// 0x0018 (0x0080 - 0x0098)
class UProductAttribute_UnlockForcedProducts_TA : public UProductAttribute_TA
{
public:
unsigned long          bUnlockTeamColor : 1;           // 0x0080 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bUnlockCustomColor : 1;         // 0x0080 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
TArray<class UProductEquipProfileSlot_ForcedProduct_TA*> SlotsToUnlock;           //
0x0088 (0x0010) [0x000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_UnlockForcedProducts_TA");
}

return uClassPointer;
};

bool ContainsSlotToUnlock(class UProductEquipProfileSlot_ForcedProduct_TA* SlotToFind);
};

// Class TAGame.ProductAttribute_UnlockMethod_TA
// 0x0001 (0x0080 - 0x0081)
class UProductAttribute_UnlockMethod_TA : public UProductAttribute_TA
{
public:
uint8_t                UnlockMethod;                 // 0x0080 (0x0001)
[0x0000000000020001] (CPF_Edit | CPF_EditConst)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_UnlockMethod_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_UserColor_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_UserColor_TA : public UProductAttribute_TA
{
public:
struct FColor           ColorValue;          // 0x0080 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_UserColor_TA");
}

return uClassPointer;
};

class FString GetOnlineProductAttributeValue();
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
static class UProductAttribute_UserColor_TA* CreateInstance(struct FColor InColorValue);
static TArray<class UProductSlot_TA*> eventGetClientAuthoritativeSlots();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& OtherAttributes);
};

// Class TAGame.ProductDatabase_Builder_TA
// 0x0004 (0x0060 - 0x0064)
class UProductDatabase_Builder_TA : public UObject
{
public:
unsigned long           bPrime : 1;          // 0x0060 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long           bClickToBuildMarked : 1;    // 0x0060 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductDatabase_Builder_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductDatabase_TA
// 0x01D4 (0x0060 - 0x0234)
class UProductDatabase_TA : public UObject
{
public:
TArray<class UProduct_TA*> Products_New; // 0x0060 (0x0010)
[0x0000000000420003] (CPF_Edit | CPF_Const | CPF_EditConst | CPF_NeedCtorLink)
TArray<class UProduct_TA*> Products_Pristine; // 0x0070 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<class UDLCPack_TA*> DLCPacks; // 0x0080 (0x0010)
[0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
TArray<class UDLCPack_TA*> DLCPacks_Pristine; // 0x0090
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<class UProductAsset_TA*> AlwaysLoadedAssets; // 0x00A0
(0x0010) [0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
TArray<struct FBodyToEngineAudioPair> BodyToEngineAudioMap; //
0x00B0 (0x0010) [0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
TArray<struct FBodyToEngineAudioPair> BodyToEngineAudioMap_Pristine; //
0x00C0 (0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
struct FGarageFolderData GarageFolderInfo; // 0x00D0 (0x0060)
[0x000000000020001] (CPF_Edit | CPF_EditConst)
struct FMap_Mirror ProductNameToProductID; // 0x0130
(0x0050) [0x0000000000001002] (CPF_Const | CPF_Native)
struct FMap_Mirror ProductNameToProductID_Pristine; // 0x0180
(0x0050) [0x0000000000003002] (CPF_Const | CPF_Native | CPF_Transient)
struct FMap_Mirror DuplicateToAssetSourceMap; // 0x01D0
(0x0050) [0x0000000000001002] (CPF_Const | CPF_Native)
TArray<struct FTemporaryProduct> TemporaryProducts; // 0x0220
(0x0010) [0x0000000000406000] (CPF_Transient | CPF_Config | CPF_NeedCtorLink)
unsigned long bUseQualityToCheckForOnlineProduct : 1; // 0x0230
(0x0004) [0x0000000000004000] [0x00000001] (CPF_Config)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.ProductDatabase_TA");
}

return uClassPointer;
};

void ResetLabels();
static bool IsOnlineUnlock(uint8_t UnlockMethod);
static class UOnlineProduct_TA* FindProductByProductID(int32_t ProductID, TArray<class UOnlineProduct_TA*>& OnlineProducts);
static class UOnlineProduct_TA* FindProductByInstanceId(struct FProductInstanceId InstanceID, TArray<class UOnlineProduct_TA*>& OnlineProducts);
static bool IsOfflineUnlock(class UProduct_TA* Product);
static bool IsDefaultProduct(class UProduct_TA* Product);
static bool IsOnlineProduct(class UProduct_TA* Product);
static bool ContainsOnlineProducts(TArray<struct FProductInstanceId>& List);
static bool StaticIsOnlineQuality(uint8_t Quality);
static bool IsOnlineID(struct FProductInstanceId ProductID);
bool IsProductIDInDLCPack(int32_t ProductID);
bool DoesPlayerOwnProduct(int32_t ProductID, unsigned long bCheckUnlockMethod, struct FUniqueNetId& PlayerID);
void RemoveIncompatibleProducts(TArray<int32_t>& OutProducts, TArray<struct FProductInstanceId>& OutOnlineProducts);
int32_t ReplaceIncompatibleProduct(int32_t ProductID, TArray<int32_t>& EquippedProducts);
bool CanEquip(int32_t ProductID, TArray<int32_t>& EquippedProducts);
void ConvertProductIDs(TArray<int32_t>& ProductIDs, TArray<class UProduct_TA*>& OutProducts);
class UProductAsset_TA* LoadAssetByName(struct FName AssetName);
static void TLoadAsset();
class UProductAsset_TA* LoadAsset(int32_t ProductID);
TArray<class UProduct_TA*> GetAllProductsBySlot(class UProductSlot_TA* Slot);
void AllProductsBySlot(class UProductSlot_TA* Slot, class UProduct_TA*& Product);
void AllProducts(class UProduct_TA*& Product);
class UProduct_TA* GetProductByName(struct FName ProductName);
class UProduct_TA* GetProductByHashID(struct FProductHashID HashID);
class UProduct_TA* GetProduct(int32_t Id);
struct FName GetProductThumbnailName(int32_t ProductID);
struct FName GetProductName(int32_t ProductID);
int32_t GetProductID(struct FName ProductName);
};

// Class TAGame.ProductEquipProfile_TA
// 0x0090 (0x0060 - 0x00F0)
class UProductEquipProfile_TA : public UObject
{
public:
    class UProductEquipProfileSlot_TA* Skin; // 0x0060 (0x0008)
    [0x000000004000001] (CPF_Edit | CPF_EditInline)
    class UProductEquipProfileSlot_TA* Wheel; // 0x0068 (0x0008)
    [0x000000004000001] (CPF_Edit | CPF_EditInline)
    class UProductEquipProfileSlot_TA* Boost; // 0x0070 (0x0008)
    [0x000000004000001] (CPF_Edit | CPF_EditInline)
    class UProductEquipProfileSlot_TA* Antenna; // 0x0078 (0x0008)
    [0x000000004000001] (CPF_Edit | CPF_EditInline)
}

```

```

class UProductEquipProfileSlot_TA*      Hat;           // 0x0080 (0x0008)
[0x0000000004000001] (CPF>Edit | CPF>EditInline)
class UProductEquipProfileSlot_TA*      PaintFinish;    // 0x0088 (0x0008)
[0x0000000004000001] (CPF>Edit | CPF>EditInline)
class UProductEquipProfileSlot_TA*      CustomFinish;   // 0x0090
(0x0008) [0x0000000004000001] (CPF>Edit | CPF>EditInline)
class UProductEquipProfileSlot_TA*      EngineAudio;    // 0x0098
(0x0008) [0x0000000004000001] (CPF>Edit | CPF>EditInline)
class UProductEquipProfileSlot_TA*      SupersonicTrail; // 0x00A0
(0x0008) [0x0000000004000001] (CPF>Edit | CPF>EditInline)
class UProductEquipProfileSlot_TA*      GoalExplosion; // 0x00A8
(0x0008) [0x0000000004000001] (CPF>Edit | CPF>EditInline)
class UProductEquipProfileSlot_TA*      PostMatchCelebration; // 0x00B0
(0x0008) [0x0000000004000001] (CPF>Edit | CPF>EditInline)

TArray<uint8_t>          LicenseWhitelist;        // 0x00B8 (0x0010)
[0x0000000004400001] (CPF>Edit | CPF>NeedCtorLink | CPF>EditInline)
TArray<uint8_t>          LicenseBlacklist;        // 0x00C8 (0x0010)
[0x0000000004400001] (CPF>Edit | CPF>NeedCtorLink | CPF>EditInline)
unsigned long             bWhitelistAll : 1;       // 0x00D8 (0x0004)
[0x0000000004000001] [0x00000001] (CPF>Edit | CPF>EditInline)
TArray<class UProductEquipProfileSlot_TA*> AllSlots; // 0x00E0
(0x0010) [0x0000004000400002] (CPF>Const | CPF>NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductEquipProfile_TA");
}

return uClassPointer;
};

int32_t __ProductEquipProfile_TA__GetForcedProducts_0x2(class UProductEquipProfileSlot_TA* FP);
bool __ProductEquipProfile_TA__GetForcedProducts_0x1(class UProductEquipProfileSlot_TA* Slot);
int32_t __ProductEquipProfile_TA__GetDefaultProducts_0x2(class UProductEquipProfileSlot_TA* FP);
bool __ProductEquipProfile_TA__GetDefaultProducts_0x1(class UProductEquipProfileSlot_TA* Slot);
TArray<int32_t> GetDefaultProducts();
TArray<int32_t> GetForcedProducts();
bool CanEquip(class UProduct_TA* InProduct);
};

// Class TAGame.ProductEquipProfileSlot_TA
// 0x0000 (0x0060 - 0x0060)
class UProductEquipProfileSlot_TA : public UObject
{
public:

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductEquipProfileSlot_TA");
}

return uClassPointer;
};

bool CanEquip(class UProduct_TA* Product);

// Class TAGame.ProductEquipProfileSlot_Custom_TA
// 0x002C (0x0060 - 0x008C)
class UProductEquipProfileSlot_Custom_TA : public UProductEquipProfileSlot_TA
{
public:
uint8_t SlotCustomization; // 0x0060 (0x0001)
[0x0000000000000001] (CPF_Edit)
TArray<class UProductAssetReference_TA*> WhitelistProducts; // 0x0068
(0x0010) [0x000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInline)
TArray<class UProductAssetReference_TA*> BlacklistProducts; // 0x0078
(0x0010) [0x000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInline)
unsigned long bUseWhitelistAsDefault : 1; // 0x0088 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductEquipProfileSlot_Custom_TA");
}

return uClassPointer;
};

int32_t GetDefaultProductID();
bool CanEquip(class UProduct_TA* Product);
};

// Class TAGame.ProductEquipProfileSlot_Disabled_TA
// 0x0000 (0x0060 - 0x0060)
class UProductEquipProfileSlot_Disabled_TA : public UProductEquipProfileSlot_TA
{
public:

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductEquipProfileSlot_Disabled_TA");
}

return uClassPointer;
};

bool CanEquip(class UProduct_TA* Product);
};

// Class TAGame.ProductEquipProfileSlot_Enabled_TA
// 0x0000 (0x0060 - 0x0060)
class UProductEquipProfileSlot_Enabled_TA : public UProductEquipProfileSlot_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductEquipProfileSlot_Enabled_TA");
}

return uClassPointer;
};

bool CanEquip(class UProduct_TA* Product);
};

// Class TAGame.ProductEquipProfileSlot_ForceProduct_TA
// 0x0008 (0x0060 - 0x0068)
class UProductEquipProfileSlot_ForceProduct_TA : public UProductEquipProfileSlot_TA
{
public:
class UProductAssetReference_TA*          ForcedProduct;           // 0x0060
(0x0008) [0x000000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class
TAGame.ProductEquipProfileSlot_ForceProduct_TA");
}

return uClassPointer;
};

int32_t GetDefaultProductID();
int32_t GetForcedProductID();
bool CanEquip(class UProduct_TA* Product);
};

// Class TAGame.ProductFilter_TA
// 0x0000 (0x0060 - 0x0060)
class UProductFilter_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductFilter_TA");
}

return uClassPointer;
};

static void AddLogLevel(uint8_t LogLevel, struct FProductFilter& Filter);
static FString GetFilterLogString(struct FProductFilter Filter);
static TArray<struct FProductHashID> GetSortedHashIDs(uint8_t SortType, class
UIdenticalProductCache_TA* IdenticalProductCache, unsigned long bUseShortName, unsigned
long bHoldXEInstance, TArray<class UOnlineProduct_TA*>& OnlineProducts, TArray<class
UProduct_TA*>& UnlockedProducts, TArray<int32_t>& FavoritedHashes, TArray<struct
FProductInstanceId>& InstanceIDs);
static TArray<struct FProductInstanceId> FilterProducts(struct FProductFilter& Filter);
static TArray<class UProduct_TA*> GetDisplayableOfflineProducts(unsigned long
bIgnoreLockedProducts, TArray<class UOnlineProduct_TA*>& OnlineProducts, TArray<int32_t>&
OfflineProducts);
};

// Class TAGame.ProductLoader_TA
// 0x0050 (0x0070 - 0x00C0)
class UProductLoader_TA : public UComponent
{
public:
TArray<class UProduct_TA*> RemainingProducts; // 0x0070
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
TArray<class UProductAsset_TA*> Assets; // 0x0080 (0x0010)
[0x0000008000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventAssetLoaded__Delegate; // 0x0090

```

```

(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventAllAssetsLoaded__Delegate;      // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductLoader_TA");
}

return uClassPointer;
};

bool __ProductLoader_TA__LoadProducts_0x1(class UProduct_TA* P);
void __ProductLoader_TA__LoadRemainingProducts_0x1(class UProduct_TA* P);
bool __ProductLoader_TA__AddEngineAudio_0x1(class UProduct_TA* P);
class UProduct_TA* __ProductLoader_TA__AddMissingSlots_0x3(class UProductSlot_TA* Slot);
class UProductSlot_TA* __ProductLoader_TA__AddMissingSlots_0x2(class UProductAsset_TA* Asset);
class UProductSlot_TA* __ProductLoader_TA__AddMissingSlots_0x1(class UProduct_TA* P);
void CancelLoad();
void OnAllAssetsLoaded();
void NotifyWhenAllAssetsLoaded(struct FScriptDelegate Callback);
void AddMissingDefaultSkinAssetForBody(class UProductAsset_Body_TA* Body, TArray<class UProductSlot_TA*> MissingSlots);
void AddMissingSlots(class UProductAsset_Body_TA* Body);
void AddEngineAudio(class UProductAsset_Body_TA* Body);
void AddForcedProducts(class UProductAsset_Body_TA* Body);
void RemoveIncompatibleProducts(class UProductAsset_Body_TA* Body);
void HandleAssetLoaded(struct FAssetLoadResult Result);
void HandleBodyLoaded(struct FAssetLoadResult Result);
class UProductAsset_TA* AllAssets(class UClass* AssetClass, class UProductAsset_TA*& Asset);
class UProductAsset_TA* GetAsset(class UClass* AssetClass);
class UProductAsset_TA* GetAssetByID(int32_t ProductID);
class UProductAsset_TA* GetAssetBySlot(class UProductSlot_TA* Slot);
void LoadClientLoadout(struct FClientLoadoutData& NewLoadout);
void LoadLoadout(struct FLoadoutData& NewLoadout);
bool ShouldLoadProduct(class UProduct_TA* Product);
void LoadRemainingProducts();
void LoadProducts(TArray<int32_t> ProductIDs);
void EventAllAssetsLoaded(class UProductLoader_TA* Loader);
void EventAssetLoaded(class UProductLoader_TA* Loader, class UProductAsset_TA* Asset);
};

// Class TAGame.ProductOverride_TA
// 0x0000 (0x0060 - 0x0060)
class UProductOverride_TA : public UObject
{
public:

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_TA");
}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);
static void ApplyOverrides(TArray<class UProductOverride_TA*> Overrides, class UObject* Target);
};

// Class TAGame.ProductOverride_MaterialParameter_TA
// 0x0050 (0x0060 - 0x00B0)
class UProductOverride_MaterialParameter_TA : public UProductOverride_TA
{
public:
class UMaterialInstanceConstant* CopyFromMIC; // 0x0060
(0x0008) [0x0000000800000001] (CPF_Edit)
struct FName MaterialName; // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FFontParameterValue> FontParameterValues; // 0x0070
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FScalarParameterValue> ScalarParameterValues; // 0x0080
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FTextureParameterValue> TextureParameterValues; // 0x0090
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FVectorParameterValue> VectorParameterValues; // 0x00A0
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_MaterialParameter_TA");
}

return uClassPointer;
};

void __ProductOverride_MaterialParameter_TA__ApplyToObject_0x1(struct
FAnimatedMaterialMesh X);
static void ForceApplyOverrideToMIC(class UProductOverride_MaterialParameter_TA* Override,
class UMaterialInstanceConstant* MIC);

```

```

void ApplyOverrideToMesh(class UMeshComponent* MeshComp);
void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductOverride_ParticleSystemParameterBase_TA
// 0x0000 (0x0060 - 0x0060)
class UProductOverride_ParticleSystemParameterBase_TA : public UProductOverride_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_ParticleSystemParameterBase_TA");
}

return uClassPointer;
};

void ApplyToParticleSystem(class UISetParameter* ParameterTarget);
void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductOverride_ParticleSystemFloatParameter_TA
// 0x001C (0x0060 - 0x007C)
class UProductOverride_ParticleSystemFloatParameter_TA : public
UProductOverride_ParticleSystemParameterBase_TA
{
public:
TArray<struct FFloatParamPair> ParameterOverrides; // 0x0060
(0x0010) [0x0000000000500001] (CPF_Edit | CPF_NeedCtorLink)
struct FFloatParamPair FloatParameter; // 0x0070 (0x000C)
[0x0000000020000000] CPF_Deprecated)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_ParticleSystemFloatParameter_TA");
}

return uClassPointer;
};

void ApplyToParticleSystem(class UISetParameter* ParameterTarget);

```

```

};

// Class TAGame.ProductOverride_ParticleSystemNameParameter_TA
// 0x0020 (0x0060 - 0x0080)
class UProductOverride_ParticleSystemNameParameter_TA : public
UProductOverride_ParticleSystemParameterBase_TA
{
public:
TArray<struct FNameParamPair> ParameterOverrides; // 0x0060
(0x0010) [0x0000000000500001] (CPF_Edit | CPF_NeedCtorLink)
struct FNameParamPair NameParameter; // 0x0070 (0x0010)
[0x0000000020000000] CPF_Deprecated)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_ParticleSystemNameParameter_TA");
}

return uClassPointer;
};

void ApplyToParticleSystem(class UISetParameter* ParameterTarget);
};

// Class TAGame.ProductOverride_ParticleSystemVectorParameter_TA
// 0x0024 (0x0060 - 0x0084)
class UProductOverride_ParticleSystemVectorParameter_TA : public
UProductOverride_ParticleSystemParameterBase_TA
{
public:
TArray<struct FVectorParamPair> ParameterOverrides; // 0x0060
(0x0010) [0x0000000000500001] (CPF_Edit | CPF_NeedCtorLink)
struct FVectorParamPair VectorParameter; // 0x0070 (0x0014)
[0x0000000020000000] CPF_Deprecated)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_ParticleSystemVectorParameter_TA");
}

return uClassPointer;
};

```

```

void ApplyToParticleSystem(class UISetParameter* ParameterTarget);
};

// Class TAGame.ProductPaint_TA
// 0x00D8 (0x0060 - 0x0138)
class UProductPaint_TA : public UObject
{
public:
class FString Label; // 0x0060 (0x0010)
[0x0000000000428003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_EditConst | CPF_NeedCtorLink)
struct FLinearColor Colors[0xC]; // 0x0070 (0x00C0)
[0x0000000000000001] (CPF_Edit)
uint8_t FinishType; // 0x0130 (0x0001)
[0x0000000000000001] (CPF_Edit)
unsigned long bVisible : 1; // 0x0134 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductPaint_TA");
}

return uClassPointer;
};

int32_t GetID();
};

// Class TAGame.ProductSlot_TA
// 0x0078 (0x0060 - 0x00D8)
class UProductSlot_TA : public UObject
{
public:
class FString Label; // 0x0060 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString PluralLabel; // 0x0070 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Description; // 0x0080 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString OnlineLabel; // 0x0090 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
int32_t SlotIndex; // 0x00A0 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
class UProduct_TA* DefaultProduct_New; // 0x00A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductTemplate_TA* ProductTemplate; // 0x00B0
(0x0008) [0x0000000804000003] (CPF_Edit | CPF_Const | CPF>EditInline)

```

```

class UProductAsset_TA*           ProductAssetTemplate;          // 0x00B8
(0x0008) [0x0000000804000003] (CPF_Edit | CPF_Const | CPF>EditInline)
class UTexture*                  Icon;                      // 0x00C0 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class UTexture*                  IconLarge;                // 0x00C8 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
unsigned long                    bReplicated : 1;        // 0x00D0 (0x0004)
[0x0000000000000003] [0x00000001] (CPF_Edit | CPF_Const)
unsigned long                    bAlwaysCooked : 1;      // 0x00D0 (0x0004)
[0x0000000000000003] [0x00000002] (CPF_Edit | CPF_Const)
unsigned long                    bDedicatedServerRelevant : 1; // 0x00D0 (0x0004)
[0x0000000000000003] [0x00000004] (CPF_Edit | CPF_Const)
unsigned long                    bAlwaysLoaded : 1;       // 0x00D0 (0x0004)
[0x0000000000000003] [0x00000008] (CPF_Edit | CPF_Const)
unsigned long                    bDefaultProductAlwaysLoaded : 1; // 0x00D0
(0x0004) [0x0000000000000003] [0x00000010] (CPF_Edit | CPF_Const)
unsigned long                    bAllowLicensedEquip : 1;    // 0x00D0 (0x0004)
[0x0000000000000003] [0x00000020] (CPF_Edit | CPF_Const)
float                           DropWeight;              // 0x00D4 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductSlot_TA");
}

return uClassPointer;
};

int32_t GetDefaultProductID();
};

// Class TAGame.ProductSpecialEdition_TA
// 0x0010 (0x0060 - 0x0070)
class UProductSpecialEdition_TA : public UObject
{
public:
class FString                 Label;                     // 0x0060 (0x0010)
[0x000000000428003] (CPF>Edit | CPF_Const | CPF>Localized | CPF>EditConst |
CPF>NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductSpecialEdition_TA");
}
}

```

```

}

return uClassPointer;
};

int32_t GetID();
};

// Class TAGame.ProductTemplate_TA
// 0x00D4 (0x0060 - 0x0134)
class UProductTemplate_TA : public UObject
{
public:
    class UProductSlot_TA*           Slot;                      // 0x0060 (0x0008)
    [0x00000000000020001] (CPF_Edit | CPF_EditConst)
    uint8_t                          UnlockMethod;            // 0x0068 (0x0001)
    [0x0000000000000000]
    uint8_t                          Quality;                 // 0x0069 (0x0001)
    [0x0000000000000000]
    uint8_t                          BaseXEStatus;          // 0x006A (0x0001)
    [0x0001000000000001] (CPF_Edit)
    class UProductAsset_Pack_TA*     Folder;                  // 0x0070 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UDLCPack_TA*              DLC;                     // 0x0078 (0x0008)
    [0x0000000000000002] (CPF_Const)
    class UClass*                   StatClass;               // 0x0080 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UProduct_TA*              RequiredProduct;        // 0x0088 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    TArray<uint8_t>                ProductIP;              // 0x0090 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<uint8_t>                AllowedList;            // 0x00A0 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    unsigned long                   bWhitelistAll : 1;      // 0x00B0 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long                   bIncludeSE : 1;         // 0x00B0 (0x0004)
    [0x0001000000000001] [0x00000002] (CPF_Edit)
    unsigned long                   bLocalized : 1;          // 0x00B0 (0x0004)
    [0x0000000800020001] [0x00000004] (CPF_Edit | CPF_EditConst)
    unsigned long                   bAutoGenerateDLCPack : 1; // 0x00B0 (0x0004)
    [0x0000000800000001] [0x00000008] (CPF_Edit)
    unsigned long                   bLicensed : 1;          // 0x00B0 (0x0004)
    [0x0000000000000000] [0x00000010]
    TArray<uint8_t>                IPWhitelist;           // 0x00B8 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<uint8_t>                IPBlacklist;            // 0x00C8 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    struct FProductReplacement2     PlatformReplacement;       // 0x00D8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<uint8_t>                PlatformExclusive;        // 0x00F0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<uint8_t>                TradeRestrictions;        // 0x0100 (0x0010)
    [0x0000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
    TArray<class UProductAttribute_TA*> Attributes;           // 0x0110 (0x0010)
}

```

```
[0x0000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF>EditInline)
class FString XELLabelKey; // 0x0120 (0x0010)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
int32_t SortPriority; // 0x0130 (0x0004)
[0x0000000000000001] (CPF>Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductTemplate_TA");
}

return uClassPointer;
};

void RecalcIPId();
class UProductAttribute_TA* GetAttribute(class UClass* AttributeClass);
};

// Class TAGame.Product_TA
// 0x0064 (0x0134 - 0x0198)
class UProduct_TA : public UProductTemplate_TA
{
public:
struct FName AssetPackageName; // 0x0138 (0x0008)
[0x0000000000000003] (CPF>Edit | CPF_Const)
class FString AssetPath; // 0x0140 (0x0010)
[0x0000000000400003] (CPF>Edit | CPF_Const | CPF_NeedCtorLink)
unsigned long bReplaced : 1; // 0x0150 (0x0004)
[0x0000000000000003] [0x00000001] (CPF>Edit | CPF_Const)
unsigned long bPack : 1; // 0x0150 (0x0004)
[0x0000000000000003] [0x00000002] (CPF>Edit | CPF_Const)
class FString Label; // 0x0158 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
class FString AsciiLabel; // 0x0168 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
class FString LongLabel; // 0x0178 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
class FString ShortAsciiLabel; // 0x0188 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Product_TA");
}
}
```

```

return uClassPointer;
};

bool IsIPAllowed(class UProduct_TA* OtherProduct);
bool CanBeArchived();
bool IsPaintable();
class FString GetDisplayLabelSlot();
uint8_t GetQuality();
bool IsContainerKey();
void SetXESpecialEditionOverride(unsigned long bNewIncludeSE);
void InternalSetBaseXEStatus(uint8_t NewStatus);
void SetBaseXEStatus(uint8_t NewStatus);
class FString GetXEDescriptionText();
class FString GetXEBlueprintDescriptionText();
class FString GetXEShopDescriptionText();
bool InternallsXETradeLocked();
bool IsXETradeLocked();
uint8_t GetFilteredXEStatus();
bool HasReactivePreview();
bool IsReactive();
bool HasPostMatchCelebration();
bool IsCurrency();
bool IsBlueprint();
bool IsContainerUnlocked();
bool CanEquip();
bool IsContainer();
bool IsSchematic();
bool IsPlatformExclusive();
bool IsLicensed();
bool IsExactlyDecryptor();
bool IsExactlyKey();
TArray<class UProductAttribute_TA*> GetAttributes();
TArray<uint8_t> GetPlatforms();
void LoadAssetAsync(struct FScriptDelegate Callback);
class UProductAsset_TA* LoadAsset();
class FString GetShortSortLabel();
class FString GetSortLabel();
class FString GetThumbnailAssetPath();
class FString GetThumbnailPackageNameForLoad();
struct FName GetThumbnailPackageName();
struct FName GetThumbnailAssetName();
class FString GetTrademarkLabel();
struct FProductHashID GetHashID();
int32_t GetID();
struct FProductHashID GetPackHashID();
};

// Class TAGame.ProductThumbnail_TA
// 0x0020 (0x0060 - 0x0080)
class UProductThumbnail_TA : public UObject
{
public:
class UTexture* Texture; // 0x0060 (0x0008)

```

```
[0x0000004000002000] (CPF_Transient)
class UTextureRenderTarget2D*           RenderTarget;          // 0x0068 (0x0008)
[0x0000004000002000] (CPF_Transient)
int32_t                                RenderCount;          // 0x0070 (0x0004)
[0x0000004000002000] (CPF_Transient)
struct FDouble                           LastRenderTime;      // 0x0078 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductThumbnail_TA");
}

return uClassPointer;
};

void ResetRenderCount();
};

// Class TAGame.ProductThumbnailAsset_TA
// 0x0018 (0x0060 - 0x0078)
class UProductThumbnailAsset_TA : public UObject
{
public:
class UTexture2D*                      Thumbnail;           // 0x0060 (0x0008)
[0x0000000000200001] (CPF_Edit)
class UTexture2D*                      ColorMaskBackground; // 0x0068 (0x0008)
[0x0000000000200000]
class UTexture2D*                      ThumbnailColorMask; // 0x0070 (0x0008)
[0x0000000000200000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductThumbnailAsset_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductThumbnailAsset_Blueprint_TA
// 0x0008 (0x0078 - 0x0080)
class UProductThumbnailAsset_Blueprint_TA : public UProductThumbnailAsset_TA
```

```

{
public:
class UTexture2D*           RevealedThumbnail;          // 0x0078 (0x0008)
[0x000300000020003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductThumbnailAsset_Blueprint_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductThumbnailQueue_TA
// 0x004C (0x0060 - 0x00AC)
class UProductThumbnailQueue_TA : public UObject
{
public:
struct FPointer           VfTable_FTickableObject;    // 0x0060 (0x0008)
[0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
TArray<int32_t>           ThumbnailSizes;           // 0x0068 (0x0010)
[0x000000000040003] (CPF>Edit | CPF_Const | CPF_NeedCtorLink)
float                      RenderDelay;              // 0x0078 (0x0004)
[0x0000000000004001] (CPF>Edit | CPF_Config)
int32_t                    RenderCountMax;         // 0x007C (0x0004)
[0x0000000000004001] (CPF>Edit | CPF_Config)
unsigned long               bDebug : 1;            // 0x0080 (0x0004)
[0x0000000000000001] [0x00000001] (CPF>Edit)
unsigned long               bCreatedThumbnailScene : 1; // 0x0080 (0x0004)
[0x0000000000002002] [0x00000002] (CPF_Const | CPF_Transient)
TArray<struct FRenderThumbnailScene>   SceneQue;        // 0x0088
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<struct FRenderThumbnailRequest> Requests;       // 0x0098
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
int32_t                     ThumbnailsCreatedSinceLastPurge; // 0x00A8 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductThumbnailQueue_TA");
}
}

```

```

return uClassPointer;
};

void CancelRenderThumbnail(class UProductAsset_TA* Asset, class UOnlineProduct_TA*
OnlineProduct, uint8_t Size);
void CancelRenderThumbnails(class UObject* Listener);
void RenderThumbnail(class UProductAsset_TA* Asset, class UOnlineProduct_TA*
OnlineProduct, uint8_t Size, int32_t TeamID, struct FScriptDelegate OnRendered);
static class UProductThumbnailQueue_TA* GetInstance();
};

// Class TAGame.ProductUtil_TA
// 0x0000 (0x0060 - 0x0060)
class UProductUtil_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductUtil_TA");
}

return uClassPointer;
};

static bool CorrectNonClientAuthoritativeAttributes(class UProductSlot_TA* Slot, TArray<class
UProductAttribute_TA*>& CheckAttributes, TArray<class UProductAttribute_TA*>&
PsyNetAttributes);
static bool CorrectDuplicateAttributes(TArray<class UProductAttribute_TA*>& CheckAttributes);
static TArray<class UOnlineProduct_TA*> FilterOutOnlineProductsWithQuality(TArray<class
UOnlineProduct_TA*>& OnlineProducts, TArray<uint8_t>& ProductQualities);
static TArray<class UOnlineProduct_TA*> GetOnlineProductsWithAttribute(class UClass*
AttributeClass, TArray<class UOnlineProduct_TA*>& OnlineProducts);
static TArray<class UOnlineProduct_TA*> GetBlueprintsWithSeriesIDs(TArray<class
UOnlineProduct_TA*>& OnlineProducts, TArray<int32_t>& SeriesIDs);
static TArray<class UOnlineProduct_TA*> GetOnlineProductsWithSeriesIDs(TArray<class
UOnlineProduct_TA*>& OnlineProducts, TArray<int32_t>& SeriesIDs);
static class UOnlineProduct_TA* FindOnlineProductWithNoAttributesFromID(int32_t ProductID,
TArray<class UOnlineProduct_TA*>& OnlineProducts, TArray<class UClass*>&
AllowedAttributes);
static TArray<class UOnlineProduct_TA*> DiffOnlineProducts(TArray<class
UOnlineProduct_TA*>& OnlineProducts, TArray<class UOnlineProduct_TA*>&
OnlineProductsToRemove);
static bool HasDuplicateOnlineProductsWithNoAttributes(int32_t ProductID, TArray<class
UOnlineProduct_TA*>& OnlineProducts, TArray<class UClass*>& AllowedAttributes);
static int32_t GetAssetSourcePID(int32_t ProductID);
static bool AreDuplicateProducts(class UProduct_TA* ProductA, class UProduct_TA* ProductB);
static void RemoveProduct(int32_t ProductID, TArray<class UOnlineProduct_TA*>&
OnlineProducts, int32_t& OutNumRemovedProducts);

```

```

static TArray<int32_t> GetAllProductIDs(TArray<class UOnlineProduct_TA*>& OnlineProducts,
TArray<int32_t>& OfflineProducts);
static TArray<class UOnlineProduct_TA*> GetOnlineProductsForSlot(class UProductSlot_TA*
Slot, TArray<class UOnlineProduct_TA*>& OnlineProducts);
static bool HasOnlineProductsForSlot(class UProductSlot_TA* Slot, TArray<class
UOnlineProduct_TA*>& OnlineProducts);
static void MapInstanceIDs(TArray<class UOnlineProduct_TA*>& OnlineProducts, TArray<struct
FProductInstanceId>& OutInstanceIDs);
static TArray<struct FProductInstanceId> RemovingMissingInstanceIDs(TArray<class
UOnlineProduct_TA*>& OnlineProducts, TArray<struct FProductInstanceId>& InstanceIDs);
static class UOnlineProduct_TA* CreateOnlineProduct(struct FOnlineProductData& ProductData);
static void ProcessOnlineData(TArray<struct FOnlineProductData>& InProductData, TArray<class
UOnlineProduct_TA*>& OutProducts);
static class UOnlineProduct_TA* GetFirstOnlineProduct(int32_t ProductID, TArray<class
UOnlineProduct_TA*>& OnlineProducts);
static int32_t GetProductIDFromOnlineID(struct FProductInstanceId InstanceID, TArray<class
UOnlineProduct_TA*>& OnlineProducts);
static class UOnlineProduct_TA* GetOnlineProductByID(struct FProductInstanceId InstanceID,
TArray<class UOnlineProduct_TA*>& OnlineProducts);
static TArray<class UOnlineProduct_TA*> GetNewProducts(TArray<struct FProductInstanceId>&
OldProductsIds, TArray<class UOnlineProduct_TA*>& NewProducts);
static bool IsOnlineID(struct FProductInstanceId ProductID);
};


```

```

// Class TAGame.PSC_Constrained_TA
// 0x0008 (0x03E0 - 0x03E8)
class UPSC_Constrained_TA : public UParticleSystemComponent
{
public:
float WorldZ; // 0x03E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bAttachedRotation : 1; // 0x03E4 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PSC_Constrained_TA");
}

return uClassPointer;
};

};

// Class TAGame.Pylon_Soccar_TA
// 0x0058 (0x04A4 - 0x04FC)
class APylon_Soccar_TA : public APylon
{
public:

```

```

struct FRotator           FieldOrientation;          // 0x04A8 (0x000C)
[0x0000000000020001] (CPF_Edit | CPF_EditConst)
struct FVector            FieldSize;              // 0x04B4 (0x000C)
[0x0000000000020001] (CPF_Edit | CPF_EditConst)
struct FVector            FieldExtent;            // 0x04C0 (0x000C)
[0x0000000000020001] (CPF_Edit | CPF_EditConst)
struct FVector            FieldCenter;             // 0x04CC (0x000C)
[0x0000000000020001] (CPF_Edit | CPF_EditConst)
TArray<class UGoal_TA*>   Goals;                  // 0x04D8 (0x0010)
[0x000000004A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component |
CPF_NeedCtorLink | CPF>EditInline)
TArray<class APlayerStart*> SpawnPoints;           // 0x04E8 (0x0010)
[0x000000000420001] (CPF_Edit | CPF_EditConst | CPF_NeedCtorLink)
float                     GroundZ;                // 0x04F8 (0x0004)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Pylon_Soccar_TA");
}

return uClassPointer;
};

struct FVector ClampPointToField(struct FVector V);
struct FVector ClampToField(struct FVector V);
void eventPostBeginPlay();

// Class TAGame.QuickChatTracker_TA
// 0x0053 (0x009D - 0x00F0)
class UQuickChatTracker_TA : public UActorComponent
{
public:
struct FMap_Mirror          GroupMessages;          // 0x00A0 (0x0050)
[0x000100000001000] (CPF_Native)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.QuickChatTracker_TA");
}

return uClassPointer;
};

```

```

void HandleChatMessage(class APlayerController_TA* PC, class FString Message, unsigned long bPreset);
void eventConstruct();
TArray<struct FQuickChatGroup> GetQuickChatGroups();
void IncrementQuickChat(class FString Key, int32_t IncrementAmount);
};

// Class TAGame.Ball_Breakout_TA
// 0x0150 (0xA48 - 0xB98)
class ABall_Breakout_TA : public ABall_TA
{
public:
TArray<struct FBreakoutDamage> DamageAtTime; // 0xA48
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<int32_t> DamageForceLevels; // 0xA58 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bClearTeamOnDamage : 1; // 0xA68 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long bCanDamageOwnTeam : 1; // 0xA68 (0x0004)
[0x0000000000000000] [0x00000002]
float MinDamageVelocity; // 0xA6C (0x0004)
[0x0000000000000000]
float MinDamageTime; // 0xA70 (0x0004)
int32_t MinImpactSpeedForCharge; // 0xA74 (0x0004)
[0x0000000000000000]
float DoubleTapTime; // 0xA78 (0x0004)
float ForceAccumDecayPerSecond; // 0xA7C (0x0004)
float ForceAccumMax; // 0xA80 (0x0004)
float ForceAccumRecent; // 0xA84 (0x0004)
[0x0000000000002000] (CPF_Transient)
uint8_t LastTeamTouch; // 0xA88 (0x0001)
[0x0000000100000020] (CPF_Net)
class ACar_TA* LastCarTouch; // 0xA90 (0x0008)
[0x0000000000000000]
int32_t LastDamage; // 0xA98 (0x0004)
[0x0000000000000000]
float LastDamageTime; // 0xA9C (0x0004)
int32_t DamagelIndex; // 0xAA0 (0x0004)
[0x0000004100000020] (CPF_Net)
float AbsorbedForce; // 0xAA4 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FAppliedBreakoutDamage AppliedDamage; // 0xAA8
(0x0018) [0x000004100000020] (CPF_Net)
struct FAccumulatedRigidBodyCollision ClosestPlatformCollision; // 0xAC0
(0x0080) [0x000004000082000] (CPF_Transient | CPF_Component)
TArray<struct FAccumulatedRigidBodyCollision> WorldCollisions; // 0xB40
(0x0010) [0x000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
struct FScriptDelegate __EventDamagelIndexChanged__Delegate; // 0xB50

```

```

(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventLastTeamTouchChanged__Delegate;      // 0x0B68 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventAppliedDamage__Delegate;          // 0x0B80 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Ball_Breakout_TA");
}

return uClassPointer;
};

void OnRBActorTeleport();
float GetDamagePercent();
bool WillDamage();
int32_t GetDamageIndexForForce(float Force);
int32_t GetDamageIndexForTime(float Time);
void eventTick(float DeltaTime);
void SetLastTeamTouch(uint8_t InLastTeamTouch);
void ForceDamageIndex(int32_t InIndex);
void SetDamageIndex(int32_t InIndex);
void ProcessClosestCollision(struct FAccumulatedRigidBodyCollision Collision);
void OnRigidBodyCollision(struct FAccumulatedRigidBodyCollision Collision);
void OnCarTouch(class ACar_TA* HitCar, uint8_t HitType);
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventAppliedDamage(class ABall_Breakout_TA* Ball, struct FAppliedBreakoutDamage Damage);
void EventLastTeamTouchChanged(class ABall_Breakout_TA* Ball, uint8_t InLastTeamTouch);
void EventDamageIndexChanged(class ABall_Breakout_TA* Ball, int32_t InDamageIndex);
};

// Class TAGame.Ball_God_TA
// 0x0088 (0x0A48 - 0x0AD0)
class ABall_God_TA : public ABall_TA
{
public:
float                      WallBounceSpeed;                         // 0x0A48 (0x0004)
[0x0001000000000002] (CPF_Const)
float                      WallBounceBlend;                        // 0x0A4C (0x0004)
[0x0001000000000002] (CPF_Const)
float                      WallBounceBackDistance;                   // 0x0A50 (0x0004)
[0x0001000000000002] (CPF_Const)
float                      TargetBlendScaleXY;                     // 0x0A54 (0x0004)
[0x0001000000000002] (CPF_Const)
float                      TargetBlendScaleZ;                      // 0x0A58 (0x0004)
[0x0001000000000002] (CPF_Const)

```

```

float TargetSpeed; // 0x0A5C (0x0004)
[0x0001000100000020] (CPF_Net)
float TargetSpeedIncrement; // 0x0A60 (0x0004)
[0x0001000000000002] (CPF_Const)
float TargetSpeedBlend; // 0x0A64 (0x0004)
[0x0001000000000002] (CPF_Const)
float MaxVelocityPitch; // 0x0A68 (0x0004)
[0x0001000000000002] (CPF_Const)
float LastSpeedIncrementTime; // 0x0A6C (0x0004)
[0x0001000000000000]
float TargetSpeedIncrementTime; // 0x0A70 (0x0004)
[0x0001000000000002] (CPF_Const)
float MinGlowBrightness; // 0x0A74 (0x0004)
[0x0001000000000002] (CPF_Const)
float MaxGlowBrightness; // 0x0A78 (0x0004)
[0x0001000000000002] (CPF_Const)
float PulseSpeedScale; // 0x0A7C (0x0004)
[0x0001000000000002] (CPF_Const)
float PulseBrightnessScale; // 0x0A80 (0x0004)
[0x0001000000000002] (CPF_Const)
float PulseMinBrightness; // 0x0A84 (0x0004)
[0x0001000000000002] (CPF_Const)
float PulseLifetime; // 0x0A88 (0x0004)
[0x0001000000000002] (CPF_Const)
struct FLinearColor GlowDefaultColor; // 0x0A8C (0x0010)
[0x0001000000000002] (CPF_Const)
struct FLinearColor MaxSpeedColor; // 0x0A9C (0x0010)
[0x0001000000000002] (CPF_Const)
struct FName AudioCurveName; // 0x0AAC (0x0008)
[0x0001000000000002] (CPF_Const)
struct FName AudiointensityName; // 0x0AB4 (0x0008)
[0x0001000000000002] (CPF_Const)
class UFXActorEvent_X* FXEvent_TeamChanged; // 0x0AC0
(0x0008) [0x0001000000000002] (CPF_Const)
class UFXActorEvent_X* FXEvent_IntensityChanged; // 0x0AC8
(0x0008) [0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Ball_God_TA");
}

return uClassPointer;
};

void __Ball_God_TA__Construct_0x2(class AGameEvent_Team_TA* _);
void __Ball_God_TA__Construct_0x1(class ABall_TA* Ball);
struct FVector GetDirectionToTarget();
struct FVector GetAimLocation();

```

```

struct FLinearColor GetMaxSpeedColor();
float eventGetTrailIntensity();
class UGoal_TA* eventGetGoalTarget();
int32_t eventGetTargetTeamIndex();
struct FLinearColor eventGetActiveTeamColor();
bool AtMaxSpeed();
float GetGlowBrightness();
float GetTargetSpeedAlpha();
void UpdatePulse();
void OnTargetSpeedChanged();
void TryIncrementTargetSpeed();
void UpdateColor();
void OnHitTeamNumChanged();
void SetCarHitTeamNum(uint8_t TeamNum);
void OnHitWorld(struct FVector HitLoc, struct FVector HitNormal, class UPhysicalMaterial* PhysMat);
void eventPostBeginPlay();
void eventConstruct();
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.Ball_Haunted_TA
// 0x0108 (0xA48 - 0xB50)
class ABall_Haunted_TA : public ABall_TA
{
public:
    struct FVector2D SeekPositionRangeLength; // 0xA48 (0x0008)
    [0x0000000000000000]
    struct FVector2D SeekPositionRangeWidth; // 0xA50 (0x0008)
    [0x0000000000000000]
    struct FVector2D SeekPositionRangeHeight; // 0xA58 (0x0008)
    [0x0000000000000000]
    struct FVector2D NextSeekTimeRange; // 0xA60 (0x0008)
    [0x0000000000000000]
    float TrappedHoverHeight; // 0xA68 (0x0004)
    [0x0000000000000000]
    float HorizontalSpeed; // 0xA6C (0x0004)
    [0x0000000000000000]
    float VerticalSpeed; // 0xA70 (0x0004)
    [0x0000000000000000]
    float ArrivalDistance; // 0xA74 (0x0004)
    [0x0000000000000000]
    float TrappedHorizontalSpeed; // 0xA78 (0x0004)
    [0x0000000000000000]
    float TrappedVerticalSpeed; // 0xA7C (0x0004)
    [0x0000000000000000]
    float TrappedCaptureTime; // 0xA80 (0x0004)
    [0x0000000000000000]
    float HitPhysicsDuration; // 0xA84 (0x0004)
    [0x0000000000000000]
    uint8_t ReplicatedBeamBrokenValue; // 0xA88 (0x0001)
    [0x0008000000000020] (CPF_Net)
    uint8_t LastTeamTouch; // 0xA89 (0x0001)
    [0x0008000000000020] (CPF_Net)

```

```

uint8_t DeactivatedGoalIndex; // 0x0A8A (0x0001)
[0x0008000000000020] (CPF_Net)
uint8_t TotalActiveBeams; // 0x0A8B (0x0001)
[0x00000000000020] (CPF_Net | CPF_Transient)
struct FVector SeekTarget; // 0x0A8C (0x000C)
[0x0000000000002000] (CPF_Transient)
float NextNeutralTime; // 0x0A98 (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long bHitPhysicsActive : 1; // 0x0A9C (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long bIsBallBeamed : 1; // 0x0A9C (0x0004)
[0x0000000000002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long bIsTrapped : 1; // 0x0A9C (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
float CurrentCaptureTime; // 0x0AA0 (0x0004)
[0x0000000000002000] (CPF_Transient)
float CaptureTimePercentage; // 0x0AA4 (0x0004)
[0x0000000000002000] (CPF_Transient)
float CaptureTimeAtExit; // 0x0AA8 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UGoal_TA* ActiveGoal; // 0xAB0 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
int32_t NumActiveBeamsByTeam[0x2]; // 0xAB8 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate __EventLastTeamTouchChanged__Delegate; // 0x0AC0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventIsTrapped__Delegate; // 0x0AD8 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventBeamIsBroken__Delegate; // 0xAF0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __ReplicatedBeamBrokenValue__ChangeNotify; // 0xB08 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __LastTeamTouch__ChangeNotify; // 0xB20 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __DeactivatedGoalIndex__ChangeNotify; // 0xB38 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Ball_Haunted_TA");
}

return uClassPointer;
};

void __Ball_Haunted_TA__PostBeginPlay_0x2();
void __Ball_Haunted_TA__PostBeginPlay_0x1();
void __Ball_Haunted_TA__SetGameEvent_0x1(class AGameEvent_Soccar_TA* SG);
void __DeactivatedGoalIndex__ChangeNotifyFunc();

```

```

void __LastTeamTouch__ChangeNotifyFunc();
void __ReplicatedBeamBrokenValue__ChangeNotifyFunc();
struct FVector GetBallDestination();
float GetVerticalVelocity(struct FVector Destination);
struct FVector GetDesiredVelocity();
void AddHauntedForces();
void TickAuthoritative();
void eventTick(float DeltaTime);
void eventOnGoalExit();
void eventOnHitGoal(class UGoal_TA* Goal, struct FVector HitLoc);
void ScoreTrapGoal();
void SetLastTeamTouch(uint8_t InLastTeamTouch);
void SetBallIsTrapped(unsigned long bValue, class UGoal_TA* Goal);
void OnHitWorld(struct FVector HitLoc, struct FVector HitNormal, class UPhysicalMaterial* PhysMat);
void SetBallHitData(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal, uint8_t HitType);
void TryBreakBeam(class ACar_TA* HitCar);
void OnCarTouch(class ACar_TA* HitCar, uint8_t HitType);
void ActivateHitPhysics();
void SetBallPhased(unsigned long bValue, int32_t TeamIndex);
void SetBallTarget();
void SetGameEvent(class AGameEvent_Soccar_TA* SoccarGame);
void eventPostBeginPlay();
void EventBeamIsBroken(class ABall_Haunted_TA* Ball);
void EventIsTrapped(class ABall_Haunted_TA* Ball, unsigned long bInIsTrapped, class UGoal_TA* Goal);
void EventLastTeamTouchChanged(class ABall_Haunted_TA* Ball, uint8_t InLastTeamTouch);
};

// Class TAGame.Ball_Trajectory_TA
// 0x0090 (0xA48 - 0xAD8)
class ABall_Trajectory_TA : public ABall_TA
{
public:
TArray<struct FVector> AccurateTrajectoryPoints; // 0xA48 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FVector> AllPoints; // 0xA58 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
float LastCalculateTime; // 0xA68 (0x0004)
[0x0000000000000000]
float FixedDeltaTime; // 0xA6C (0x0004)
float TotalPathDistance; // 0xA70 (0x0004)
int32_t FrameCount; // 0xA74 (0x0004)
int32_t TotalStepCount; // 0xA78 (0x0004)
int32_t TotalFrameCount; // 0xA7C (0x0004)
int32_t CurrentStepCount; // 0xA80 (0x0004)
int32_t StepToPutAPointAt; // 0xA84 (0x0004)

```

```

[0x0000000000000000]
struct FVector
[0x0000000000000000]
struct FVector
[0x0000000000000000]
struct FVector
[0x0000000000000000]
struct FVector
[0x0000000000000000]
struct FRotator
[0x0000000000000000]
struct FTrajectorySetup
[0x0000000000000000]
class UBallTrajectoryComponent_TA*           TrajectoryComponentToUpdate;      //
0x0AD0 (0x0008) [0x000000004080008] (CPF_EditObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Ball_Trajectory_TA");
}

return uClassPointer;
};

bool CalculateTrajectory(struct FTrajectorySetup Settings, float TrajectoryScale, class
UBallTrajectoryComponent_TA* InTrajectoryComponent, struct FVector StartLocation, struct
FRotator StartRotation, struct FVector LaunchVelocity, struct FVector InAngularVelocity, unsigned
long bForceUpdate);
bool CanEverShowTrajectory();
void eventPostBeginPlay();
};

// Class TAGame.Vehicle_TA
// 0x0100 (0x07A8 - 0x08A8)
class AVehicle_TA : public ARBActor_TA
{
public:
class UCarMeshComponent_TA*           CarMesh;                  // 0x07A8
(0x0008) [0x00000000408000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_Component | 
CPF_EditInline)
class UVehicleSim_TA*                VehicleSim;             // 0x07B0 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FStickyForceData              StickyForce;            // 0x07B8 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FAutoFlipData                 AutoFlip;               // 0x07C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                      bDriving : 1;          // 0x07C8 (0x0004)
[0x0000004000002020] [0x00000001] (CPF_Net | CPF_Transient)
unsigned long                      bReplicatedHandbrake : 1; // 0x07C8 (0x0004)
[0x0000000000000002] [0x00000002] (CPF_Const | CPF_Net | CPF_Transient)
unsigned long                      bJumped : 1;           // 0x07C8 (0x0004)

```

```

[0x0000008000002000] [0x00000004] (CPF_Transient)
unsigned long bDoubleJumped : 1; // 0x07C8 (0x0004)
[0x0000008000002000] [0x00000008] (CPF_Transient)
unsigned long bOnGround : 1; // 0x07C8 (0x0004)
[0x0000004000002000] [0x00000010] (CPF_Transient)
unsigned long bSuperSonic : 1; // 0x07C8 (0x0004)
[0x0000004000002000] [0x00000020] (CPF_Transient)
unsigned long bPodiumMode : 1; // 0x07C8 (0x0004)
[0x000000100002020] [0x00000040] (CPF_Net | CPF_Transient)
unsigned long bHasPostMatchCelebration : 1; // 0x07C8 (0x0004)
[0x0000008000002020] [0x00000080] (CPF_Net | CPF_Transient)
struct FVehicleInputs Input; // 0x07CC (0x0020)
[0x00000000000002002] (CPF_Const | CPF_Transient)
uint8_t ReplicatedThrottle; // 0x07EC (0x0001)
[0x00000000000002022] (CPF_Const | CPF_Net | CPF_Transient)
uint8_t ReplicatedSteer; // 0x07ED (0x0001)
[0x00000000000002022] (CPF_Const | CPF_Net | CPF_Transient)
class AAIController_TA* AIController; // 0x07F0 (0x0008)
[0x0000004000002000] (CPF_Transient)
class APlayerController_TA* PlayerController; // 0x07F8 (0x0008)
[0x0000004000002000] (CPF_Transient)
class APRI_TA* PRI; // 0x0800 (0x0008)
[0x0000004000002000] (CPF_Transient)
int32_t VehicleUpdateTag; // 0x0808 (0x0004)
[0x000000800002002] (CPF_Const | CPF_Transient)
struct FCarInteractionData CarInteraction; // 0x0810 (0x0010)
[0x0000000000000000]
struct FVector LocalCollisionOffset; // 0x0820 (0x000C)
[0x00000000000002002] (CPF_Const | CPF_Transient)
struct FVector LocalCollisionExtent; // 0x082C (0x000C)
[0x00000000000002002] (CPF_Const | CPF_Transient)
int32_t LastBallTouchFrame; // 0x0838 (0x0004)
[0x00000000000002000] (CPF_Transient)
int32_t LastBallImpactFrame; // 0x083C (0x0004)
[0x00000000000002000] (CPF_Transient)
class ACarComponent_Boost_TA* BoostComponent; // 0x0840
(0x0008) [0x00000000000002000] (CPF_Transient)
class ACarComponent_Dodge_TA* DodgeComponent; // 0x0848
(0x0008) [0x00000000000002000] (CPF_Transient)
class ACarComponent_AirControl_TA* AirControlComponent; // 0x0850
(0x0008) [0x00000000000002000] (CPF_Transient)
class ACarComponent_Jump_TA* JumpComponent; // 0x0858
(0x0008) [0x00000000000002000] (CPF_Transient)
class ACarComponent_DoubleJump_TA* DoubleJumpComponent; //
0x0860 (0x0008) [0x00000000000002000] (CPF_Transient)
int32_t PodiumSpot; // 0x0868 (0x0004)
[0x0000008000002020] (CPF_Net | CPF_Transient)
int32_t PMCAAnimIdx; // 0x086C (0x0004)
[0x0000008000002020] (CPF_Net | CPF_Transient)
class UPitchTekDrawingComponent_TA* PitchTekComponent; // 0x0870
(0x0008) [0x000000000408000A] (CPF_Const | CPF_ExportObject | CPF_Component | CPF_EditInline)
class ULocalPlayerAudioParamsComponent_TA*
LocalPlayerAudioParamsComponent; // 0x0878 (0x0008) [0x0000000004082008]

```

```
(CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float TimeBelowSupersonicSpeed; // 0x0880 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UNetworkConfig_TA* NetworkConfig; // 0x0888 (0x0008)
[0x0000800000000000]
struct FScriptDelegate __EventPRIChanged__Delegate; // 0x0890
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Vehicle_TA");
}

return uClassPointer;
};

bool eventAllowCarComponentActivate(class ACarComponent_TA* CarComponent);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void DrawCollisionBox(struct FVector ActorLocation, struct FRotator ActorRotation, struct FColor BoxColor, unsigned long bPersistent);
void DumpVehicleSetupForBalboa();
void DumpVehicleSetup();
void ForceNetPacketIfNearBall();
void eventVehicleSettingsUpdated();
void eventOnSuperSonicChanged();
void eventOnGroundChanged();
bool eventCanLand();
bool IsHitLocationWithinForwardAngle(float YawAngleDegrees, float PitchAngleDegrees, struct FTimeOfImpactData& Impact);
bool IsCarHitAngleWithinForwardAngleCurve(class AVehicle_TA* OtherCar, struct FTimeOfImpactData& Impact, struct FInterpCurveFloat& YawAngleDegreesCurve, struct FInterpCurveFloat& PitchAngleDegreesCurve);
bool IsCarHitAngleWithinForwardAngle(class AVehicle_TA* OtherCar, float YawAngleDegrees, float PitchAngleDegrees, struct FTimeOfImpactData& Impact);
bool IsCarWithinForwardEllipticalCone(class AVehicle_TA* OtherCar, float YawAngleDegrees, float PitchAngleDegrees, struct FTimeOfImpactData& Impact);
float GetForwardSpeed();
float GetTimeOffGround();
float GetTimeOnGround();
struct FVector GetGroundNormal();
bool IsOnWall();
bool IsOnGround();
int32_t GetNumWheelWorldContacts();
int32_t GetNumWheelContacts();
void ZeroMovementVariables();
void eventSetVehicleInput(struct FVehicleInputs NewInput);
void SetAudioPMCAAnimIdx();
void SetAudioPodiumSpot();
void SetPodiumSpot(int32_t inSpot);
```

```

void EnablePodiumMode();
void SetDriving(unsigned long bDrive);
void InitAudioParams();
void OnPRIChanged();
void OnControllerChanged();
void UnPossessed();
void PossessedBy(class AController* C);
void eventReplicatedEvent(struct FName VarName);
void EventPRIChanged(class AVehicle_TA* Vehicle);
};

// Class TAGame.RBHistory_TA
// 0x0030 (0x0070 - 0x00A0)
class URBHistory_TA : public UComponent
{
public:
TArray<int32_t> FrameSnapshots; // 0x0070 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<unsigned long> ServerSnapshots; // 0x0080 (0x0010)
[0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
TArray<struct FRBPhysicsSnapshot> RBPhysicsSnapshots; // 0x0090
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RBHistory_TA");
}

return uClassPointer;
};

};

// Class TAGame.RBVehicleHistory_TA
// 0x0048 (0x00A0 - 0x00E8)
class URBVehicleHistory_TA : public URBHistory_TA
{
public:
float InputFadeTime; // 0x00A0 (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FRBVehicleSnapshot> RBVehicleSnapshots; // 0x00A8
(0x0010) [0x0000000000482002] (CPF_Const | CPF_Transient | CPF_Component |
CPF_NeedCtorLink)
TArray<struct FCarComponentHistory> ComponentHistories; // 0x00B8
(0x0010) [0x0000000000402002] (CPF_Const | CPF_Transient | CPF_NeedCtorLink)
struct FVehicleInputs BackupVehicleInputs; // 0x00C8 (0x0020)
[0x0000000000002002] (CPF_Const | CPF_Transient)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RBVehicleHistory_TA");
}

return uClassPointer;
};

};

// Class TAGame.Replay_TA
// 0x0280 (0x0060 - 0x02E0)
class UReplay_TA : public UObject
{
public:
class FString           ReplayName;          // 0x0060 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                 EngineVersion;       // 0x0070 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
int32_t                 LicenseeVersion;     // 0x0074 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
int32_t                 NetVersion;          // 0x0078 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
int32_t                 ReplayVersion;       // 0x007C (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t                 ReplayLastSaveVersion; // 0x0080 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t                 GameVersion;         // 0x0084 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t                 BuildID;             // 0x0088 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t                 Changelist;         // 0x008C (0x0004)
[0x0000000000000002] (CPF_Const)
class FString           BuildVersion;        // 0x0090 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
int32_t                 ReserveMegabytes;    // 0x00A0 (0x0004)
[0x0000000000004000] (CPF_Config)
float                  RecordFPS;           // 0x00A4 (0x0004)
[0x0000000040004002] (CPF_Const | CPF_Config | CPF_EditInlineNotify)
float                  KeyframeDelay;        // 0x00A8 (0x0004)
[0x0000000000004002] (CPF_Const | CPF_Config)
int32_t                 MaxChannels;         // 0x00AC (0x0004)
[0x0000000000004002] (CPF_Const | CPF_Config)
int32_t                 MaxReplaySizeMB;     // 0x00B0 (0x0004)
[0x0000000000004002] (CPF_Const | CPF_Config)
TArray<class UClass*>   PlaybackClasses;      // 0x00B8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UClass*>   IgnoreClasses;        // 0x00C8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString           FilePath;            // 0x00D8 (0x0010)

```

[0x0000000040402000] (CPF\_Transient | CPF\_NeedCtorLink | CPF\_EditInlineNotify)  
class FString Id; // 0x00E8 (0x0010)  
[0x0000000040400000] (CPF\_NeedCtorLink | CPF\_EditInlineNotify)  
struct FName MapName; // 0x00F8 (0x0008)  
[0x0000000040000002] (CPF\_Const | CPF\_EditInlineNotify)  
class FString Date; // 0x0100 (0x0010)  
[0x0000000040400002] (CPF\_Const | CPF\_NeedCtorLink | CPF\_EditInlineNotify)  
int32\_t NumFrames; // 0x0110 (0x0004)  
[0x0000000040000002] (CPF\_Const | CPF\_EditInlineNotify)  
struct FName MatchType; // 0x0114 (0x0008)  
[0x0000000040000000] (CPF>EditInlineNotify)  
class FString PlayerName; // 0x0120 (0x0010)  
[0x0000000040400000] (CPF\_NeedCtorLink | CPF>EditInlineNotify)  
unsigned long bFileCorrupted : 1; // 0x0130 (0x0004)  
[0x0000000040000002] [0x00000001] (CPF\_Const | CPF>EditInlineNotify)  
unsigned long bForceKeyframe : 1; // 0x0130 (0x0004)  
[0x00000000000002002] [0x00000002] (CPF\_Const | CPF\_Transient)  
unsigned long bLoadedNetPackages : 1; // 0x0130 (0x0004)  
[0x00000000000002002] [0x00000004] (CPF\_Const | CPF\_Transient)  
unsigned long bDebug : 1; // 0x0130 (0x0004)  
[0x00000000000002001] [0x00000008] (CPF>Edit | CPF\_Transient)  
TArray<uint8\_t> Data; // 0x0138 (0x0010)  
[0x00000000000402002] (CPF\_Const | CPF\_Transient | CPF\_NeedCtorLink)  
TArray<struct FReplayKeyframe> KeyFrames; // 0x0148 (0x0010)  
[0x00000000000402002] (CPF\_Const | CPF\_Transient | CPF\_NeedCtorLink)  
uint8\_t ReplayState; // 0x0158 (0x0001)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
int32\_t CurrentFrame; // 0x015C (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
int32\_t NextKeyframe; // 0x0160 (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
float CurrentTime; // 0x0164 (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
TArray<struct FReplayActorChannel> Channels; // 0x0168 (0x0010)  
[0x00000000000402002] (CPF\_Const | CPF\_Transient | CPF\_NeedCtorLink)  
struct FMap\_Mirror ChannelsMap; // 0x0178 (0x0050)  
[0x0000000000003002] (CPF\_Const | CPF\_Native | CPF\_Transient)  
class UPackageMap\* PackageMap; // 0x01C8 (0x0008)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
TArray<struct FName> Levels; // 0x01D0 (0x0010)  
[0x00000000000402002] (CPF\_Const | CPF\_Transient | CPF\_NeedCtorLink)  
TArray<int32\_t> FreeChannels; // 0x01E0 (0x0010)  
[0x00000000000402002] (CPF\_Const | CPF\_Transient | CPF\_NeedCtorLink)  
float AccumulatedDeltaTime; // 0x01F0 (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
float TimeToSkipTo; // 0x01F4 (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
int32\_t FrameToSkipTo; // 0x01F8 (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
int32\_t PlayersOnlyTicks; // 0x01FC (0x0004)  
[0x00000000000002002] (CPF\_Const | CPF\_Transient)  
struct FPointer WriterAr; // 0x0200 (0x0008)  
[0x0000000000003002] (CPF\_Const | CPF\_Native | CPF\_Transient)  
struct FPointer ReaderAr; // 0x0208 (0x0008)

```

[0x0000000000003002] (CPF_Const | CPF_Native | CPF_Transient)
TArray<struct FReplayLogItem> Logs; // 0x0210 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FTimelineKeyframe> TimelineKeyframes; // 0x0220
(0x0010) [0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class UCameraTrack_TA* CameraTrack; // 0x0230 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate __EventPlaybackStopped__Delegate; // 0x0238
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSpawned__Delegate; // 0x0250
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPreTimeSkip__Delegate; // 0x0268
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPostTimeSkip__Delegate; // 0x0280
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPlayedFrame__Delegate; // 0x0298
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventMissingObject__Delegate; // 0x02B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPlaybackError__Delegate; // 0x02C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Replay_TA");
}

return uClassPointer;
};

void eventReplayPlaybackError();
float GetPlaybackTime();
static bool IsFromBeforeGameVersion(uint8_t BeforeGameVersion);
static bool IsFromBeforeReplayVersion(uint8_t BeforeReplayVersion);
struct FName GetMapToLoad();
void SetReplayName(class FString NewName);
void RemoveTimelineKeyframe(int32_t KeyframeIndex);
void AddTimelineKeyframe(struct FName Type, int32_t frame);
static void ReplayLog(class UObject* Context, class FString Text);
void eventTrimData(int32_t FirstKeyframe, int32_t FirstFrame);
class UReplay_TA* CreateCopy(float StartTime);
void ImportReplay(class FString Path);
void ExportReplay(class FString Path);
void SerializeReplay(TArray<uint8_t>& OutBytes);
void SkipToFrame(int32_t frame, unsigned long bFlush);
void SkipToTime(float Time, unsigned long bFlush);
class AActor* GetPlaybackActor(struct FName RecordedActorName);
float GetReplayTimeSeconds();
void StopPlayback();

```

```

void StartPlaybackAtFrame(int32_t StartFrame);
void StartPlaybackAtTimeSeconds(float StartTime);
void StopRecord();
void StartRecord();
void Tick(float DeltaTime);
void AllReplayActors(class UClass* BaseClass, class AActor*& Actor);
void eventPreExport();
void EventPlaybackError(class UReplay_TA* Replay);
void EventMissingObject(class UReplay_TA* Replay, class FString ObjectPath);
void EventPlayedFrame(class UReplay_TA* Replay);
void EventPostTimeSkip(class UReplay_TA* Replay);
void EventPreTimeSkip(class UReplay_TA* Replay);
void EventSpawned(class UReplay_TA* Replay, class AActor* A);
void EventPlaybackStopped(class UReplay_TA* Replay);
};

// Class TAGame.ReplayCompatibilityActor_TA
// 0x0010 (0x0268 - 0x0278)
class AReplayCompatibilityActor_TA : public AActor
{
public:
TArray<struct FReplayRenamedActor> RenamedActors; // 0x0268
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ReplayCompatibilityActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.ReplayComponent_TA
// 0x0007 (0x009D - 0x00A4)
class UReplayComponent_TA : public UActorComponent
{
public:
unsigned long bRecording : 1; // 0x00A0 (0x0004)
[0x0000000000002002] [0x00000001] (CPF_Const | CPF_Transient)
unsigned long bPlaying : 1; // 0x00A0 (0x0004)
[0x0000000000002002] [0x00000002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ReplayComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.ReplayManager_TA
// 0x00E8 (0x0060 - 0x0148)
class UReplayManager_TA : public UObject
{
public:
class FString           ReplaysPath;          // 0x0060 (0x0010)
[0x0000000000404003] (CPF_Edit | CPF_Const | CPF_Config | CPF_NeedCtorLink)
class FString           EpicReplaysPath;      // 0x0070 (0x0010)
[0x0000000000404003] (CPF_Edit | CPF_Const | CPF_Config | CPF_NeedCtorLink)
TArray<struct FPointer> ExportTasks;          // 0x0080 (0x0010)
[0x0000000000003000] (CPF_Native | CPF_Transient)
TArray<struct FPointer> ClassTasks;          // 0x0090 (0x0010)
[0x0000000000003000] (CPF_Native | CPF_Transient)
TArray<struct FPointer> ImportTasks;          // 0x00A0 (0x0010)
[0x0000000000003000] (CPF_Native | CPF_Transient)
struct FPointer         HeadersCountTask;     // 0x00B0 (0x0008)
[0x0000000000003000] (CPF_Native | CPF_Transient)
TArray<struct FPointer> HeaderLoadTasks;      // 0x00B8 (0x0010)
[0x0000000000003000] (CPF_Native | CPF_Transient)
TArray<struct FReplayExportCallbackData> ExportCallbacks;    // 0x00C8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FReplayImportCallbackData> ImportCallbacks;   // 0x00D8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate  _EventPreLoadReplay__Delegate; // 0x00E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate  _EventExportFinished__Delegate; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate  _EventImportFinished__Delegate; // 0x0118
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate  _EventHeadersLoaded__Delegate; // 0x0130
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ReplayManager_TA");
}

return uClassPointer;
};

```

```

void PlayReplay(class UReplay_TA* Replay, class FString Map, class FString URL);
void PlayReplayFile(class FString FilePath, class FString Map, class FString URL);
void PlayReplayID(class FString ReplayID, class FString Map, class FString URL);
void eventSetTick(unsigned long bTick);
void Tick(float DeltaTime);
bool IsReplayFilenameTaken(class FString Filename);
static bool IsValidFilename(class FString Filename);
void DeleteReplayFile(class FString ReplayID, unsigned long bCheckEditorOnlyData);
void ExportReplayBytes(class FString ReplayID, struct FScriptDelegate Callback,
TArray<uint8_t>& Bytes);
void ExportReplay(class UReplay_TA* Replay, struct FScriptDelegate Callback);
void ImportReplayFile(class FString FilePath, struct FScriptDelegate Callback, unsigned long
bSynchronous);
void LoadHeaderFile(class FString FilePath, struct FScriptDelegate Callback);
void StartAsyncHeadersCountTask(class FString ReplaysFolderPath);
void LoadOtherPCPlatformHeaders(struct FScriptDelegate Callback);
void LoadHeaders(struct FScriptDelegate Callback);
class FString GetReplayFolder();
void Init();
static class UReplayManager_TA* GetInstance();
bool IsExporting();
void EventHeadersLoaded(class UReplayManager_TA* Manager, TArray<struct
FReplayHeaderLoadResult> Headers, class UError* Error);
void EventImportFinished(class UReplayManager_TA* Manager, class UReplay_TA* Replay, class
UError* Error);
void EventExportFinished(class UReplayManager_TA* Manager, class FString Id, class UError*
Error);
void EventPreLoadReplay(class UReplayManager_TA* Manager, class UReplay_TA* Replay);
};

// Class TAGame.RLBot_SessionRecorder_TA
// 0x0030 (0x0268 - 0x0298)
class ARLBot_SessionRecorder_TA : public AActor
{
public:
int32_t MaxRecordSizeMB; // 0x0268 (0x0004)
[0x0001000000004001] (CPF_Edit | CPF_Config)
TArray<uint8_t> Data; // 0x0270 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FPointer Archive; // 0x0280 (0x0008)
[0x0001000000001000] (CPF_Native)
TArray<struct FPlayerActorIDPair> PlayerActorIDMap; // 0x0288
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RLBot_SessionRecorder_TA");
}

```

```

return uClassPointer;
};

void eventDestroyed();
void StopRecording();
void StartRecording();
};

// Class TAGame.RLBot_Util_TA
// 0x0028 (0x0060 - 0x0088)
class URLBot_Util_TA : public UObject
{
public:
class FString ControllerURL; // 0x0060 (0x0010)
[0x0001000000404000] (CPF_Config | CPF_NeedCtorLink)
class FString OutputFilePath; // 0x0070 (0x0010)
[0x0001000000404000] (CPF_Config | CPF_NeedCtorLink)
float PacketSendRate; // 0x0080 (0x0004)
[0x000100000004000] (CPF_Config)
unsigned long bRecordInput : 1; // 0x0084 (0x0004)
[0x0001000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RLBot_Util_TA");
}

return uClassPointer;
};

void __RLBot_Util_TA__Init_0x1(class FString _);
class APlayerReplicationInfo* eventGetSpectatorViewTarget();
struct FPlayerStats eventGetPlayerStats(class APlayerReplicationInfo* PlayerReplicationInfo);
void eventSetPaused(unsigned long bPaused);
void eventEndMatch(class AGameEvent_TA* GameEvent);
void HandleReplayFrame(class UReplay_TA* Replay);
void eventReplayFinished();
void ExportReplay(class FString Path);
void HandleStatEvent(class APRI_TA* Receiver, class APRI_TA* Victim, class UStatEvent_TA* StatEvent);
void HandleHitBall(class ACar_TA* Car, class ABall_TA* Ball, struct FVector HitLocation, struct FVector HitNormal);
class AAIController_TA* eventSpawnCustomBot(class AGameEvent_TA* G, class FString BotName, uint8_t Team, struct FBotLoadout Loadout);
class AAIController_TA* eventSpawnClassicBot(class AGameEvent_TA* G, class FString BotName, uint8_t Team, float Skill, struct FBotLoadout Loadout);
class AAIController_TA* SpawnBot(class AGameEvent_TA* G, class FString BotName, uint8_t TeamNum, class UProductAsset_Bot_TA* Asset, struct FBotLoadout Loadout);

```

```

class UProductAsset_Bot_TA* GetBotAsset(class FString BotName);
int32_t eventGetTeamScore(class AGameEvent_TA* G, int32_t Team);
int32_t eventGetDisplaySeconds(class AGameEvent_TA* G);
bool eventGetOvertime(class AGameEvent_TA* G);
bool eventGetBallHasBeenHit(class AGameEvent_TA* G);
void PreLoadMap();
void PostLoadMap();
void eventStopMatch();
void eventStartMatch();
void eventInit();
static class URLBot_Util_TA* GetInstance();
};

// Class TAGame.SampleHistory_TA
// 0x0070 (0x0070 - 0x00E0)
class USampleHistory_TA : public UComponent
{
public:
    class USampleRecordSettings_TA*           RecordSettings;          // 0x0070
    (0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    class FString                           Title;                  // 0x0078 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<struct FGraphSummaryMethod>       Summaries;             // 0x0088
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    float                                 YMin;                 // 0x0098 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                                 YMax;                 // 0x009C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                                 GoodValue;            // 0x00A0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                                 BadValue;              // 0x00A4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                                 BaseValue;            // 0x00A8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    TArray<struct FSample>                Samples;               // 0x00B0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t                               SampleIndex;           // 0x00C0 (0x0004)
    [0x0000000000000000]
    float                                 AccumTime;            // 0x00C4 (0x0004)
    [0x0000000000000000]
    struct FSample                         PendingSample;        // 0x00C8 (0x0008)
    [0x0000000000000000]
    unsigned long                          bHasPendingSample : 1; // 0x00D0 (0x0004)
    [0x0000000000000000] [0x00000001]
    class UStatGraphDrawer_TA*            Drawer;                // 0x00D8 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.SampleHistory_TA");
}

return uClassPointer;
};

void Tick(float DeltaTime);
void AddSample(float NewValue);
float GetSummaryValue(uint8_t Type, float MaxSampleAge, unsigned long bAbsoluteValue);
class USampleHistory_TA* SetBaseValue(float InBaseValue);
class USampleHistory_TA* SetGoodBadValues(float InGoodValue, float InBadValue);
class USampleHistory_TA* SetGraphMaxMin(float MaxValue, float MinValue);
class USampleHistory_TA* AddSummary(uint8_t Type, float MaxSampleAge, unsigned long bAbsValue);
class USampleHistory_TA* SetTitle(class FString InTitle);
};

// Class TAGame.SampleRecordSettings_TA
// 0x0008 (0x0070 - 0x0078)
class USampleRecordSettings_TA : public UComponent
{
public:
float MaxSampleAge; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RecordRate; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SampleRecordSettings_TA");
}

return uClassPointer;
};

};

// Class TAGame.SaveGameManager_TA
// 0x00A0 (0x0060 - 0x0100)
class USaveGameManager_TA : public UObject
{
public:
class USaveObjectManager_TA* SaveObjectManager; // 0x0060
(0x0008) [0x0000040000000000]
class FString SaveDataID; // 0x0068 (0x0010)
[0x000004000400000] (CPF_NeedCtorLink)
class ULocalPlayer_TA* Player; // 0x0078 (0x0008)
[0x000004000002000] (CPF_Transient)

```

```

class USaveData_TA*           SaveData;                  // 0x0080 (0x0008)
[0x0000004004400008] (CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
class UObjectProvider*        SaveObjectProvider;      // 0x0088 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float                         MinTimeBetweenSaves;    // 0x0090 (0x0004)
[0x0000000000004000] (CPF_Config)
float                         LastSaveTime;            // 0x0094 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UError*                 DataLoadError;          // 0x0098 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UError*                 DataSaveError;          // 0x00A0 (0x0008)
[0x0000004000002000] (CPF_Transient)
unsigned long                 bDataLoaded : 1;       // 0x00A8 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long                 bUserIgnoredLoadFailure : 1; // 0x00A8 (0x0004)
[0x0000004000002000] [0x00000002] (CPF_Transient)
unsigned long                 bEnableSaving : 1;      // 0x00A8 (0x0004)
[0x0000004000002000] [0x00000004] (CPF_Transient)
unsigned long                 bDisableSaveOSS : 1;     // 0x00A8 (0x0004)
[0x0000004000002000] [0x00000008] (CPF_Transient)
unsigned long                 bDirty : 1;             // 0x00A8 (0x0004)
[0x0000000000002000] [0x00000010] (CPF_Transient)
class FString                SaveFolderPath;          // 0x00B0 (0x0010)
[0x0000000000404000] (CPF_Config | CPF_NeedCtorLink)
class FString                EpicSaveFolderPath;      // 0x00C0 (0x0010)
[0x0000000000404000] (CPF_Config | CPF_NeedCtorLink)
struct FScriptDelegate        __EventDataLoaded__Delegate; // 0x00D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate        __EventDataSaved__Delegate;   // 0x00E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SaveGameManager_TA");
}

return uClassPointer;
};

bool DeleteMount();
void HandleNoSpaceSystemDialogComplete(unsigned long bContinueWithouSaving);
void Save_Internal();
void Save_Timer();
void HandleDataSaved(struct FSaveObjectResult Result);
void ForceSave();
void Save();
bool CanSave();
void UserIgnoredLoadFailure(unsigned long bValue);
void HandleProceedWithDefaultSaveData();

```

```

void DisableSaving();
void EnableSaving();
void HandleDataLoaded(struct FLoadObjectResult Result);
void OnLocalSaveDataDisabled();
void Load();
void NotifyWhenLoaded(struct FScriptDelegate Callback);
void HandlePlayerLoginStatusChange(class UOnlinePlayer_X* OnlinePlayer);
void SetSaveDataID(class FString InSaveDataID);
void Init(class ULocalPlayer_TA* InPlayer);
void eventConstruct();
class FString GetSaveFileName(class FString SaveFolder);
class FString GetSaveFolder();
class FString GetSaveEnvFolder();
void EventDataSaved(class USaveGameManager_TA* Manager, class USaveData_TA*
MySaveData, class UError* Error);
void EventDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*
MySaveData, class UError* Error);
};

// Class TAGame.SaveObjectManager_TA
// 0x00F0 (0x0060 - 0x0150)
class USaveObjectManager_TA : public UObject
{
public:
    struct FPointer           VfTable_FTickableObject;           // 0x0060 (0x0008)
    [0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
    TArray<struct FPointer>   ImportTasks;                     // 0x0068 (0x0010)
    [0x0000000000103000] (CPF_Native | CPF_Transient)
    TArray<struct FScriptDelegate> ImportCallbacks;           // 0x0078 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FPointer>   ExportTasks;                     // 0x0088 (0x0010)
    [0x0000000000103000] (CPF_Native | CPF_Transient)
    TArray<struct FPointer>   ListFilesTasks;                 // 0x0098 (0x0010)
    [0x0000000000103000] (CPF_Native | CPF_Transient)
    TArray<struct FPointer>   DeleteTasks;                     // 0x00A8 (0x0010)
    [0x0000000000103000] (CPF_Native | CPF_Transient)
    TArray<struct FScriptDelegate> ListFilesCallbacks;         // 0x00B8 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    int32_t                  KeepNewestSaveFileCount;          // 0x00C8 (0x0004)
    [0x0000000000000000]
    class UObject*>          SaveData;                        // 0x00D0 (0x0008)
    [0x0000000000000000]
    class FString             SavePath;                        // 0x00D8 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t                  ControllerId;                   // 0x00E8 (0x0004)
    [0x0000000000000000]
    unsigned long             bSaving : 1;                    // 0x00EC (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    unsigned long             bPlayerSaveData : 1;            // 0x00EC (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    struct FScriptDelegate    __EventDataLoaded__Delegate;     // 0x00F0
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate    __EventDataSaved__Delegate;      // 0x0108
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```
struct FScriptDelegate           __OnFilesListed__Delegate;           // 0x0120 (0x0018)
[0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventFilesDeleted__Delegate;       // 0x0138
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SaveObjectManager_TA");
}

return uClassPointer;
};

void CancelGetHeaders(struct FScriptDelegate Callback);
void GetHeaders(int32_t InControllerID, class FString InPath, class FString InPattern, struct FScriptDelegate Callback);
void SetNewFileSaveCount(int32_t Count);
void eventHandleSaveResult(uint8_t Result);
class UError* GetSaveError(uint8_t Result);
class UError* GetLoadError(uint8_t InResult);
void LoadAsync(int32_t ControllerIndex, class UObject* RootObj, class FString PathName,
unsigned long bExactFileMatch, struct FScriptDelegate Callback);
struct FLoadObjectResult Load(int32_t InControllerID, class UObject* RootObj, class FString PathName,
unsigned long bExactFileMatch);
void SaveAsync(int32_t InControllerID, class UObject* RootObj, class FString PathName,
unsigned long bExactFileMatch);
struct FSaveObjectResult Save(int32_t InControllerID, class UObject* RootObj, class FString PathName,
unsigned long bExactFileMatch);
void DecrementSaveManagerCounter();
void IncrementSaveManagerCounter();
static int32_t GetSaveDataSpace(int32_t ControllerIndex);
class FString GetPlatformSavePath(class FString DefaultPath, struct FUniqueNetId PlayerID);
bool DeleteMount(int32_t ControllerIndex, class FString PathName);
void DeleteFilesW(int32_t ControllerIndex, class FString PathName, int32_t
KeepNewestItemCount);
void DeleteFilesAsyncW(int32_t ControllerIndex, class FString PathName, int32_t
KeepNewestItemCount);
bool IsReadingSaveData();
void LoadGameDataAsync(int32_t ControllerIndex, class UObject* RootObj, class FString
PathName, unsigned long bExactFileMatch, struct FScriptDelegate Callback);
uint8_t LoadGameData(int32_t ControllerIndex, class UObject* RootObj, class FString PathName,
unsigned long bExactFileMatch);
bool IsDoneSaving();
bool IsSaving();
uint8_t UpdateSaveResult(int32_t ControllerIndex);
uint8_t SaveGameDataAsync(int32_t ControllerIndex, class UObject* RootObj, class FString
PathName, unsigned long bExactFileMatch);
uint8_t SaveGameData(int32_t ControllerIndex, class UObject* RootObj, class FString PathName,
unsigned long bExactFileMatch);
```

```

void EventFilesDeleted(int32_t ControllerIndex, class FString PathName);
void OnFilesListed(TArray<class FString> Files, uint8_t Result);
void EventDataSaved(struct FSaveObjectResult Result);
void EventDataLoaded(struct FLoadObjectResult Result);
};

// Class TAGame.SaveObjectSerializer_TA
// 0x0005 (0x0060 - 0x0065)
class USaveObjectSerializer_TA : public USerializer_X
{
public:
int32_t Version; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
uint8_t Result; // 0x0064 (0x0001)
[0x0000000000002002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SaveObjectSerializer_TA");
}

return uClassPointer;
};

class USaveObjectSerializer_TA* SetVersion(int32_t InVersion);
void DeserializeObject(class UObject* Target, TArray<uint8_t>& InData);
void SerializeObject(class UObject* Target, TArray<uint8_t>& OutData);
static class USaveObjectSerializer_TA* CreateNew(int32_t InVersion);
};

// Class TAGame.SeqAct_AnalyzeMusicFromCsv_TA
// 0x0024 (0x0160 - 0x0184)
class USeqAct_AnalyzeMusicFromCsv_TA : public USequenceAction
{
public:
struct FPointer VfTable_FTickableObject; // 0x0160 (0x0008)
[0x0000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
class FString CsvFileName; // 0x0168 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
float StartOverrideTime; // 0x0178 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float OverrideTime; // 0x017C (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long bTickable : 1; // 0x0180 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_AnalyzeMusicFromCsv_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqAct_CarMatinee_TA
// 0x0028 (0x0288 - 0x02B0)
class USeqAct_CarMatinee_TA : public USeqAct_Interp
{
public:
class FString CinematicCarArchPath; // 0x0288 (0x0010)
[0x0000080000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
unsigned long bHideAfterPause : 1; // 0x0298 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bHideAfterCompleted : 1; // 0x0298 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
TArray<class UClass*> ClassesToNotShow; // 0x02A0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_CarMatinee_TA");
}

return uClassPointer;
};

class AActor* eventCreateCars(class ASkeletalMeshActorMAT_Products_TA* SkelMeshActor);
void UpdateLinkedVariablesToNewCar(class AActor* Car, class
ASkeletalMeshActorMAT_Products_TA* SkelMeshActor);
void UpdateAttachmentsToNewCar(class AActor* Car, class
ASkeletalMeshActorMAT_Products_TA* SkelMeshActor);
};

// Class TAGame.SeqAct_DumpMovie_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_DumpMovie_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_DumpMovie_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqAct_HideDuplicateCar_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_HideDuplicateCar_TA : public USequenceAction
{
public:
class FString ActorNameToHide; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bKeepAllInstancesHidden : 1; // 0x0170 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bFoundDemoedCar : 1; // 0x0170 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_HideDuplicateCar_TA");
}

return uClassPointer;
};

void __SeqAct_HideDuplicateCar_TA__Activated_0x1(struct FReplayActorChannel X);
void CheckDemo(class ACar_TA* Victim, struct FDemolishData Data);
void CheckCar(class ACar_TA* Car);
void eventCheckCarToHide(class AVehicle_TA* CarToCheck);
void eventActivated();
};

// Class TAGame.SeqAct_SetLoadout_TA
// 0x00A0 (0x0160 - 0x0200)
class USeqAct_SetLoadout_TA : public USequenceAction
{
public:
TArray<class UProductAsset_TA*> Products; // 0x0160 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductWithSettings> ProductsWithPaint; // 0x0170
(0x0010) [0x00000000000400001] (CPF_Edit | CPF_NeedCtorLink)

```

```

class UProductAsset_PaintFinish_TA*      TeamFinish;           // 0x0180
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductAsset_PaintFinish_TA*      AccentFinish;        // 0x0188
(0x0008) [0x0000000000000001] (CPF_Edit)
unsigned long               bUseInGameColors : 1;          // 0x0190 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long               bOverwriteCurrent : 1;         // 0x0190 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long               bUseDefaultLoadout : 1;       // 0x0190 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long               bApplyToAllCars : 1;          // 0x0190 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
struct FLinearColor          TeamColor;            // 0x0194 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor          AccentColor;          // 0x01A4 (0x0010)
[0x0000000000000001] (CPF_Edit)
uint8_t                     ForcedTeamOverride;    // 0x01B4 (0x0001)
[0x0000000000000001] (CPF_Edit)
class UProductPaint_TA*      BodyPaint;            // 0x01B8 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      SkinPaint;            // 0x01C0 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      WheelPaint;          // 0x01C8 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      BoostPaint;          // 0x01D0 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      SSTrailPaint;        // 0x01D8 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      AntennaPaint;        // 0x01E0 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      HatPaint;             // 0x01E8 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*      GoalExplosionPaint; // 0x01F0 (0x0008)
[0x0000000000000000]
class ACar_TA*                ParentCar;           // 0x01F8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetLoadout_TA");
}

return uClassPointer;
};

bool __SeqAct_SetLoadout_TA__GetLoadout_0x1(class UProductAttribute_TA* CurAttr);
static int32_t eventGetObjClassVersion();
void eventApplyLoadoutToSkeletalMesh(class ASkeletalMeshActorMAT_Products_TA* Target);
void HandleOverrideLoadout(class ACar_TA* Car);

```

```

class UProductAsset_TA* GetProductAsset(class UClass* ClassToFind, class UObject* PackageToCompare);
void ApplyChangesToProductAsset(class UCarMeshComponent_TA* CarMesh, class UProductPaint_TA* PaintToApply, class UObject* ObjectToChange, class UClass* ProductAssetClass, class UAssetAttribute_TeamEdition_TA* TeamEdition, int32_t TeamID, class UObject* PackageToCompare);
void CheckAndApplyChangesToComponent(class UObject* ObjectToCheck, class UObject* ObjectToCompare, class UProductPaint_TA* Paint, class UAssetAttribute_TeamEdition_TA* TeamEdition, int32_t TeamID);
void ApplyChangesToAttachment(struct FProductAttachment CurrentAttachment, class UProductPaint_TA* Paint, class UAssetAttribute_TeamEdition_TA* TeamEdition, int32_t TeamID);
void ApplyNewChangesToProductAssets(class UProductLoader_TA* ProductLoader);
int32_t GetTeamIndex(class ACar_TA* Car);
void HandleFirstLoadout(class ACar_TA* Car);
struct FClientLoadoutData GetLoadout(class ACar_TA* Car, class UProductAsset_Skin_TA*& OutSkinAsset);
static class FString GetPlayerNameFromCar(class ACar_TA* Car);
void CopyPropertiesFromCar(class ACar_TA* Car);
void eventActivated();
};

// Class TAGame.SeqAct_SetReplayFX_TA
// 0x002C (0x0160 - 0x018C)
class USeqAct_SetReplayFX_TA : public USequenceAction
{
public:
    float FocusDistance; // 0x0160 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t FocusBlur; // 0x0164 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float FocusStrength; // 0x0168 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    unsigned long bUseGreenScreen : 1; // 0x016C (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long bUseVignetteOverlay : 1; // 0x016C (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    struct FLinearColor GreenScreenColor; // 0x0170 (0x0010)
    [0x0000000000000001] (CPF_Edit)
    uint8_t ImageFilterIndex; // 0x0180 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    float ImageFilterIntensity; // 0x0184 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float Vignette; // 0x0188 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetReplayFX_TA");
        }
    }
};

```

```

return uClassPointer;
};

void SetSystemSettings();
void InitValuesFromSystemSettings();
void eventActivated();
void eventCopySettingsFromPostProcessManager();
};

// Class TAGame.SeqAct_SetTimeDilation_TA
// 0x0040 (0x0160 - 0x01A0)
class USeqAct_SetTimeDilation_TA : public USequenceAction
{
public:
struct FPointer           VfTable_FTickableObject;          // 0x0160 (0x0008)
[0x00000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
unsigned long              bUseCurve : 1;                  // 0x0168 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long              bActivated : 1;                 // 0x0168 (0x0004)
[0x0000000000000000] [0x00000002]
float                      CurrentTime;                // 0x016C (0x0004)
[0x0000000000000000]
float                      EndTime;                   // 0x0170 (0x0004)
[0x0000000000000000]
float                      SlomoValue;                 // 0x0174 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FRawDistributionFloat SlomoCurve;                // 0x0178 (0x0028)
[0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetTimeDilation_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqAct_SpawnFXActor_TA
// 0x0080 (0x0160 - 0x01E0)
class USeqAct_SpawnFXActor_TA : public USequenceAction
{
public:
struct FPointer           VfTable_FTickableObject;          // 0x0160 (0x0008)
[0x00000000000801002] (CPF_Const | CPF_Native | CPF_NoExport)
class AFXActor_X*          FXActor;                    // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit)

```

```

class UProductAssetReference_TA*          Product;           // 0x0170 (0x0008)
[0x0000000004000001] (CPF_Edit | CPF_EditInline)
class AActor*                           Transform;        // 0x0178 (0x0008)
[0x0000000000000000]
struct FVector                         Location;        // 0x0180 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator                         Rotation;        // 0x018C (0x000C)
[0x0000000000000001] (CPF_Edit)
class AActor*                           ActorToSpawnOn; // 0x0198 (0x0008)
[0x0000000000000000]
class AFXActor_X*                      SpawnedFXActor; // 0x01A0 (0x0008)
[0x0000000000000000]
class UProductPaint_TA*                Paint;           // 0x01A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                            PaintParameterNames; // 0x01B0 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                          bTickable : 1; // 0x01B8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
class UMaterialInstanceConstant*       BallLocMic;     // 0x01C0 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate                 _EventSpawned__Delegate; // 0x01C8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SpawnFXActor_TA");
}

return uClassPointer;
};

void SpawnFXActor(struct FVector LocationToSpawnOn, struct FRotator RotationToSpawnWith);
static void GetFXActorAndPaintSettingsFromProduct(class UProductAssetReference_TA*
ProductAssetRef, unsigned long bWillBePainted, class AFXActor_X*& OutFXActor, class
UProductAttribute_PaintSettings_TA*& OutPaintSettings);
void eventActivated();
void EventSpawned(class USeqAct_SpawnFXActor_TA* SpawnFXActor);
};

// Class TAGame.SeqEvent_ReplayTime_TA
// 0x0008 (0x017C - 0x0184)
class USeqEvent_ReplayTime_TA : public USequenceEvent
{
public:
float                           TimeSeconds;      // 0x0180 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_ReplayTime_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_UIStateChange_TA
// 0x0005 (0x017C - 0x0181)
class USeqEvent_UIStateChange_TA : public USequenceEvent
{
public:
uint8_t UIState; // 0x0180 (0x0001)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_UIStateChange_TA");
}

return uClassPointer;
};

};

// Class TAGame.SimpleSpringComponent_TA
// 0x0078 (0x0070 - 0x00E8)
class USimpleSpringComponent_TA : public UComponent
{
public:
struct FSimpleSpringSettings Spring; // 0x0070 (0x0050)
[0x0000000000000001] (CPF_Edit)
unsigned long bEnabled : 1; // 0x00C0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bInitialized : 1; // 0x00C0 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
struct FVector Displacement; // 0x00C4 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector MassLocation; // 0x00D0 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector MassVelocity; // 0x00DC (0x000C)
[0x0000000000002000] (CPF_Transient)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SimpleSpringComponent_TA");
}

return uClassPointer;
};

void eventDrawDebug(struct FRotator BaseRotation);
struct FVector GetNormalizedDisplacement();
void AddForce(struct FVector Force);
void UpdateSpring(float DeltaTime, struct FVector& BaseLocation, struct FRotator& BaseRotation);
void UpdatePrimitive(float DeltaTime, class UPrimitiveComponent* BaseComponent);
};

// Class TAGame.AntennaComponent_TA
// 0x0078 (0x00E8 - 0x0160)
class UAntennaComponent_TA : public USimpleSpringComponent_TA
{
public:
struct FName TopperSocket; // 0x00E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class USkeletalMesh* AntennaMesh; // 0x00F0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float AntennaSegmentExtension; // 0x00F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UMaterialInterface* AntennaMaterial; // 0x0100 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAnimTree* AnimTreeTemplate; // 0x0108 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FName> BendyBoneControllers; // 0x0110 (0x0010)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float BendScale; // 0x0120 (0x0004)
[0x0000000000000000]
float PitchScale; // 0x0124 (0x0004)
[0x0000000000000001] (CPF_Edit)
float RollScale; // 0x0128 (0x0004)
[0x0000000000000001] (CPF_Edit)
float PitchZScale; // 0x012C (0x0004)
[0x0000000000000001] (CPF_Edit)
float RollZScale; // 0x0130 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant* TopperMaterial; // 0x0138
(0x0008) [0x0000000000002000] (CPF_Transient)
float LastTopperWindSpeedParamValue; // 0x0140 (0x0004)
[0x0000000000002000] (CPF_Transient)
class USkeletalMeshComponent_TA* AntennaSK; // 0x0148
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | )

```

```

CPF>EditInline)
TArray<class USkelControlSingleBone*>      BendyControls;          // 0x0150
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AntennaComponent_TA");
}

return uClassPointer;
};

void UpdateSpring(float DeltaTime, struct FVector& BaseLocation, struct FRotator&
BaseRotation);
void UpdatePrimitive(float DeltaTime, class UPrimitiveComponent* BaseComponent);
void SetTopperMaterial(class UMaterialInstanceConstant* InTopperMaterial);
void InitAntenna(class USkeletalMeshComponent_TA* AttachToMesh);
};

// Class TAGame.SkelControlRotationSpring_TA
// 0x005C (0x0130 - 0x018C)
class USkelControlRotationSpring_TA : public USkelControlSingleBone
{
public:
struct FRotator           RotationMin;          // 0x0130 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator           RotationMax;         // 0x013C (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector            SpringToPitch;        // 0x0148 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector            SpringToYaw;          // 0x0154 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector            SpringToRoll;         // 0x0160 (0x000C)
[0x0000000000000001] (CPF_Edit)
class USimpleSpringComponent_TA*   SpringComponent; // 0x0170
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF>EditInline)
struct FVector           TestSpringForce;       // 0x0178 (0x000C)
[0x0000000800000001] (CPF_Edit)
float                   TestSpringDelay;        // 0x0184 (0x0004)
[0x0000000800000001] (CPF_Edit)
float                   LastTestSpringTime;     // 0x0188 (0x0004)
[0x0000000800002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkelControlRotationSpring_TA");
}

return uClassPointer;
};

};

// Class TAGame.SkelControlSingleBoneCopy_TA
// 0x0078 (0x0130 - 0x01A8)
class USkelControlSingleBoneCopy_TA : public USkelControlSingleBone
{
public:
struct FName ControlToCopyName; // 0x0130 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FAxisConversion TranslationX; // 0x0138 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FAxisConversion TranslationY; // 0x0144 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FAxisConversion TranslationZ; // 0x0150 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FAxisConversion RotationPitch; // 0x015C (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FAxisConversion RotationYaw; // 0x0168 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FAxisConversion RotationRoll; // 0x0174 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector TranslationScale; // 0x0180 (0x000C)
[0x0000000000000000]
struct FVector RotationScale; // 0x018C (0x000C)
[0x0000000000000000]
unsigned long bConvertedScales : 1; // 0x0198 (0x0004)
[0x0000000000000000] [0x00000001]
class USkelControlSingleBone* ControlToCopy; // 0x01A0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkelControlSingleBoneCopy_TA");
}

return uClassPointer;
};

};

// Class TAGame.SkelControlSingleBoneParameterized_TA

```

```

// 0x00F0 (0x0130 - 0x0220)
class USkelControlSingleBoneParameterized_TA : public USkelControlSingleBone
{
public:
    struct FRawDistributionFloat          Pitch;           // 0x0130 (0x0028)
    [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
    struct FRawDistributionFloat          Yaw;            // 0x0158 (0x0028)
    [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
    struct FRawDistributionFloat          Roll;           // 0x0180 (0x0028)
    [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
    struct FRawDistributionFloat          PitchSpeed;     // 0x01A8 (0x0028)
    [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
    struct FRawDistributionFloat          YawSpeed;       // 0x01D0 (0x0028)
    [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
    struct FRawDistributionFloat          RollSpeed;      // 0x01F8 (0x0028)
    [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SkelControlSingleBoneParameterized_TA");
        }

        return uClassPointer;
    };
};

// Class TAGame.SkelControlTether_TA
// 0x0010 (0x0100 - 0x0110)
class USkelControlTether_TA : public USkelControlBase
{
public:
    struct FName                  AnchorBoneName;      // 0x0100 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float                         DistanceMin;        // 0x0108 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                         DistanceMax;        // 0x010C (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SkelControlTether_TA");
        }

        return uClassPointer;
    };
};

```

```

return uClassPointer;
};

};

// Class TAGame.SkeletalMeshActor_NoSlomo_TA
// 0x0004 (0x02CC - 0x02D0)
class ASkeletalMeshActor_NoSlomo_TA : public ASkeletalMeshActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkeletalMeshActor_NoSlomo_TA");
}

return uClassPointer;
};

};

// Class TAGame.SkeletalMeshActorMAT_Products_TA
// 0x0108 (0x02E0 - 0x03E8)
class ASkeletalMeshActorMAT_Products_TA : public ASkeletalMeshActorMAT
{
public:
class UProductAsset_Body_TA*           BodyAsset;           // 0x02E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_Skin_TA*           SkinAsset;          // 0x02E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_PaintFinish_TA*    TeamFinishAsset;    // 0x02F0
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductAsset_PaintFinish_TA*    CustomFinishAsset; // 0x02F8
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductAsset_Wheel_TA*          WheelAsset;         // 0x0300
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductAsset_Boost_TA*          BoostAsset;        // 0x0308 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_Attachment_TA*     AntennaAsset;      // 0x0310
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductAsset_Attachment_TA*     HatAsset;          // 0x0318
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductPaint_TA*               BodyPaint;         // 0x0320 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductPaint_TA*               SkinPaint;         // 0x0328 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductPaint_TA*               WheelPaint;        // 0x0330 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                      bInitBoostFX : 1;   // 0x0338 (0x0004)

```

```

[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bUseInGameColors : 1; // 0x0338 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
struct FLinearColor TeamColor; // 0x033C (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor CustomColor; // 0x034C (0x0010)
[0x0000000000000001] (CPF_Edit)
float FakeBoostVelocity; // 0x035C (0x0004)
[0x0000000200000001] (CPF_Edit)
float DeactivateBoostVelocity; // 0x0360 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UCarPreviewMeshComponent_TA* CarMeshComp; // 0x0368
(0x0008) [0x000000024080008] (CPF_ExportObject | CPF_Component | CPF>EditInline | CPF_Deprecated)
class USkeletalMeshComponent* FLWheelSKComponent; // 0x0370
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class USkeletalMeshComponent* FRWheelSKComponent; // 0x0378
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class USkeletalMeshComponent* BLWheelSKComponent; // 0x0380
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class USkeletalMeshComponent* BRWheelSKComponent; // 0x0388
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class UStaticMeshComponent* FLWheelSMComponent; // 0x0390
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class UStaticMeshComponent* FRWheelSMComponent; // 0x0398
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class UStaticMeshComponent* BLWheelSMComponent; // 0x03A0
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class UStaticMeshComponent* BRWheelSMComponent; // 0x03A8
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
TArray<class UMeshComponent*> WheelMeshComponents; // 0x03B0
(0x0010) [0x0000000004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF>EditInline)
class UCarPreviewMeshComponent_TA* PreviewMesh; // 0x03C0
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
TArray<struct FBothWheelAssets> AdditionalWheels; // 0x03C8
(0x0010) [0x0000000000480000] (CPF_Component | CPF_NeedCtorLink)
class UNameplateComponent_TA* NameplateComponent; // 0x03D8
(0x0008) [0x0000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UProductLoader_TA* ProductLoader; // 0x03E0 (0x0008)
[0x0000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkeletalMeshActorMAT_Products_TA");
}

return uClassPointer;
};

void AddProduct(class UProductAsset_TA* InProduct, struct FClientLoadoutData& OutLoadout);
struct FClientLoadoutData GetClientLoadoutData();
void HandleAllProductsLoaded(class UProductLoader_TA* InProductLoader);
void UpdateWithLoadout(struct FLoadoutData Loadout);
void CreateAttachments();
void UpdateBoost();
void eventCreateBoostActor();
void UpdateAllComponents();
void eventHandleBeingReplacedByCar(class AActor* ReplacementActor);
void ApplyPaintForProductAsset(class UCarMeshComponentBase_TA* CarMesh, class
UProductAsset_TA* ProductAsset, class UProductPaint_TA* PaintToApply, class UObject*
ObjectToChange);
void eventApplyProductPaints();
void eventTick(float DeltaTime);
void SetDefaultAssets();
void eventPostBeginPlay();
void ConditionalCreateMeshes();
};

// Class TAGame.SkeletalMeshComponent_NoSlomo_TA
// 0x0000 (0x07B0 - 0x07B0)
class USkeletalMeshComponent_NoSlomo_TA : public USkeletalMeshComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkeletalMeshComponent_NoSlomo_TA");
}

return uClassPointer;
};

};

// Class TAGame.SkeletalMeshComponent_TA
// 0x0008 (0x07B0 - 0x07B8)
class USkeletalMeshComponent_TA : public USkeletalMeshComponent

```

```

{
public:
float FOVOverride; // 0x07B0 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
unsigned long bIgnoreScriptAttach : 1; // 0x07B4 (0x0004)
[0x0000000000002002] [0x00000001] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkeletalMeshComponent_TA");
}

return uClassPointer;
};

class UAnimNodeSequence* FindAnimNodeSequence(struct FName InSeqName);
void PrintDebugInfo(class UDebugDrawer* Drawer);
class UParticleSystemComponent* CreateAttachmentPSC(class UParticleSystem* PS, class
UObject* OuterOverride);
class USkeletalMeshComponent_TA* CreateAttachmentSK(class USkeletalMesh* Mesh, class
UMaterialInterface* MaterialOverride, class UObject* OuterOverride);
class UStaticMeshComponent_TA* CreateAttachmentSMC(class UStaticMesh* Mesh, class
UMaterialInterface* MaterialOverride, unsigned long bCreateNewMIC, class UObject*
OuterOverride);
void ModifyAttachmentPrimitive(class UPrimitiveComponent* PrimComp);
void eventTick(float DeltaTime);
void eventOnDetached();
void eventOnAttached();
};

// Class TAGame.CarMeshComponentBase_TA
// 0x0118 (0x07B8 - 0x08D0)
class UCarMeshComponentBase_TA : public USkeletalMeshComponent_TA
{
public:
class UProductAsset_Body_TA* BodyAsset; // 0x07B8 (0x0008)
[0x0000000000000000]
class UProductAsset_Skin_TA* SkinAsset; // 0x07C0 (0x0008)
[0x0000000000000000]
class UProductAsset_Wheel_TA* WheelAsset; // 0x07C8
(0x0008) [0x0000000000000000]
struct FLinearColor TeamColorOverride; // 0x07D0 (0x0010)
[0x0000004000002000] (CPF_Transient)
struct FLinearColor CustomColorOverride; // 0x07E0 (0x0010)
[0x0000004000002000] (CPF_Transient)
int32_t Team; // 0x07F0 (0x0004)
[0x0000004000002000] (CPF_Transient)
int32_t TeamColorID; // 0x07F4 (0x0004)
[0x0000004000002000] (CPF_Transient)

```

```

int32_t          CustomColorID;           // 0x07F8 (0x0004)
[0x0000004000002000] (CPF_Transient)
class UProductAsset_Logo_TA*      Logo;           // 0x0800 (0x0008)
[0x0000004000002000] (CPF_Transient)
unsigned long      bSwapLogoColors : 1;        // 0x0808 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long      bBoosting : 1;            // 0x0808 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long      bLocalPlayer : 1;         // 0x0808 (0x0004)
[0x0000004000000000] [0x00000004]
class UProductAsset_PaintFinish_TA*   TeamFinish;    // 0x0810
(0x0008) [0x0000004000002000] (CPF_Transient)
class UProductAsset_PaintFinish_TA*   CustomFinish;  // 0x0818
(0x0008) [0x0000004000002000] (CPF_Transient)
struct FClubColorSet      ClubColors;     // 0x0820 (0x0008)
[0x0000004000002000] (CPF_Transient)
TArray<class UAntennaComponent_TA*>   Antennas;      // 0x0828
(0x0010) [0x0000008004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
TArray<class UMeshComponent*>      WheelMeshes;    // 0x0838
(0x0010) [0x0000008004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
class AFXActor_X*      BodyFX;        // 0x0848 (0x0008)
[0x0000000000002000] (CPF_Transient)
TArray<class AFXActor_X*>      FXActors;      // 0x0850 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UAttachmentBehavior_TA*> Behaviors;    // 0x0860
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class AFXActor_X*      BodyFXActor;   // 0x0870 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FClientLoadoutOnlineData      ProductAttributes; // 0x0878
(0x0010) [0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FName>      ProductAttachmentSockets; // 0x0888
(0x0010) [0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
int32_t          QueuedTeamFinishID; // 0x0898 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t          QueuedCustomFinishID; // 0x089C (0x0004)
[0x0000000000002000] (CPF_Transient)
class USkeletalMeshComponent_SplitBody_TA* PMCSplitBodyMesh; // 0x08A0 (0x0008) [0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<struct FInheritedObjects> ObjectsWithInheritedSettings; // 0x08A8
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate      _EventPaintSettingsUpdated__Delegate; // 0x08B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarMeshComponentBase_TA");
}
}

```

```

return uClassPointer;
};

void __CarMeshComponentBase_TA__AddVisualAsset_0x1(struct FProductAttachment A);
void __CarMeshComponentBase_TA__InitBodyVisuals_0x2(struct FProductAttachment A);
void __CarMeshComponentBase_TA__InitBodyVisuals_0x1(struct FProductAttachment A);
void __CarMeshComponentBase_TA__CreateWheelMeshes_0x1(class UMeshComponent* W);
void __CarMeshComponentBase_TA__ApplyPaintSettings_0x2(class UMeshComponent* W);
void eventApplyColorsToCar(struct FLinearColor TeamColor, struct FLinearColor CustomColor);
void ApplyPaintSettingsToObject(class UProductPaint_TA* ProductPaint, class
UProductAttribute_PaintSettings_TA* PaintSettings, class UObject* ObjectToPaint, class
UProductAsset_TA* Asset);
void eventApplyPaintToObject(class UProductPaint_TA* ProductPaint, class
UProductAttribute_PaintSettings_TA* PaintSettings, class UObject* ObjectToPaint, class
UProductAsset_TA* Product);
void SetWheelMesh(class UMeshComponent* MeshComp, class UProductAsset_Wheel_TA*
Asset, uint8_t WheelPos, struct FName BoneName, struct FVehicleAxeSettings& Axle);
struct FVector CalculatePresetSuspensionOffset();
class AFXActor_X* AttachFXActor(struct FProductAttachment& AttachStruct);
class UAttachmentBehavior_TA* AddBehavior(class UProductAsset_TA* Asset, class
UActorComponent* Component, class UAttachmentBehavior_TA* BehaviorArchetype);
void AddBehaviors(class UProductAsset_TA* Asset, class UActorComponent* Component,
TArray<class UAttachmentBehavior_TA*>& BehaviorArchetypes);
void AddCarInheritedObject(class UProductAsset_TA* Asset, class UObject* ObjectToAdd, class
UProductAttribute_InheritCarSetting_TA* InheritSettings);
void UpdateInheritedObjects();
void AttachOrnament(class UProductAsset_TA* Asset, struct FProductAttachment&
AttachStruct);
bool CanPaintAntennaMesh(class UProductAsset_TA* Asset, class UAntennaComponent_TA*
Antenna, uint8_t Socket);
void SetMaterialColorParams(class UMaterialInstanceConstant* MatInst, struct FLinearColor
PaintColor, struct FName ColorParam, struct FName FullColorParam);
bool CanCorrectColor(unsigned long bTeamColor);
void SetMaterialColors(class UMaterialInstanceConstant* MatInst);
struct FLinearColor GetCorrectedPaintColor(struct FLinearColor InColor);
void SetMeshMaterialColors(class UMeshComponent* Mesh);
void CreateWheelMICs();
void SetPaintFinishParametersToAccent(class UMaterialInstanceConstant* MatInst, unsigned
long blsPaintFinishDisable, class UProductAsset_PaintFinish_TA* PaintFinishToApply, struct
FParameterInformation OptionalPaintFinishToApply, class FString Prefix);
void SetPaintFinishParameters(class UMeshComponent* Mesh);
void ApplyPaintSettings();
void OnPaintChanged();
static void ResetMaterials(class UMeshComponent* Mesh);
void InitMaterials();
bool FilterTeamAttributes(class UProductAttribute_TA* Attribute);
void RemoveProductAttributeFromProduct(class UClass* AttributeClassToRemove, int32_t
ProductSlot);
void ApplyAttributes(class UProductAsset_TA* Asset, class UObject* Target, TArray<class
UClass*>& AttributesToIgnore);
TArray<class UProductAttribute_TA*> GetAttributes(class UProduct_TA* Product);
void SetIsLocalPlayer(unsigned long blsLocal);
struct FLinearColor GetCustomColor();

```

```

struct FLinearColor GetTeamColor();
void SetLogo(class UProductAsset_Logo_TA* InLogo, unsigned long bSwapColors);
void SetCustomFinish(class UProductAsset_PaintFinish_TA* Finish);
void HandleCustomFinishLoaded(struct FAssetLoadResult Result);
void SetCustomFinishID(int32_t ProductID);
void SetCustomColorOverride(struct FLinearColor NewColor);
void SetCustomColorID(int32_t ColorID);
void SetTeamFinish(class UProductAsset_PaintFinish_TA* Finish);
void HandleTeamFinishLoaded(struct FAssetLoadResult Result);
void SetTeamFinishID(int32_t ProductID);
void SetClubColors(struct FClubColorSet Colors);
void SetTeamColorOverride(struct FLinearColor NewColor);
void SetTeamColorID(int32_t InTeam, int32_t InColorID);
void ClearColorOverrides();
void SetTeamPaint(struct FLoadoutTeamPaint Paint);
class UPrimitiveComponent* eventCreateWheelAttachment(class UProductAsset_Wheel_TA*
Asset, class UMeshComponent* WheelMesh, struct FWheelAttachment InAttachment);
void CreateWheelMesh(class UProductAsset_Wheel_TA* Asset, uint8_t WheelPos, struct
FVehicleAxeSettings Axle, struct FName BoneName);
void CreateWheelMeshes();
class UProductAsset_Wheel_TA* GetAxeWheelAsset(class
UAssetAttribute_ForceWheelAxe_TA* ForceWheelAxe, class UProductAsset_Wheel_TA*
InWheelAsset, uint8_t Axe);
void SetMaterialParameters(int32_t ElementIdx, struct FMaterialParams& Params);
void eventOnDetached();
void InitBodyVisuals();
class UPhysicsAsset* GetPhysicsAsset();
void InitBodyAsset();
void AddVisualAsset(class UProductAsset_TA* Asset);
void SetBodyAsset(class UProductAsset_Body_TA* Body);
void eventInitAssets();
void EventPaintSettingsUpdated(class UCarMeshComponentBase_TA* CarMeshComponent);
};

// Class TAGame.CarMeshComponent_TA
// 0x0080 (0x08D0 - 0x0950)
class UCarMeshComponent_TA : public UCarMeshComponentBase_TA
{
public:
float BoostBlendInTime; // 0x08D0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostBlendOutTime; // 0x08D4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostDriveFlapScale; // 0x08D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float WheelSuspensionBlendRate; // 0x08DC (0x0004)
[0x0000000000000001] (CPF_Edit)
float WheelSteerBlendRate; // 0x08E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FWheelSkelControlSet> WheelControls; // 0x08E8
(0x0010) [0x0000004000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
float BoostFlyHeight; // 0x08F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<class USkelControlBase*> BoostFlyControls; // 0x0900

```

```

(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class USimpleSpringComponent_TA*           ChassisSpringComponent;          // 0x0910 (0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class USkelControlSingleBone*             ChassisSpringControl;          // 0x0918 (0x0008) [0x0000004000002000] (CPF_Transient)
TArray<class USkelControlBase*>          BoostControls;                // 0x0920 (0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long                           bInAir : 1;                  // 0x0930 (0x0004) [0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long                           bBoostFlying : 1;              // 0x0930 (0x0004) [0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long                           bPreviewSupersonic : 1;        // 0x0930 (0x0004) [0x0000000000002000] [0x00000004] (CPF_Transient)
unsigned long                           bWantsBoostCameraAdjustment : 1; // 0x0930 (0x0004) (0x0004) [0x0000000000002000] [0x00000008] (CPF_Transient)
float                                 Throttle;                   // 0x0934 (0x0004) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate                 __EventAttached__Delegate;       // 0x0938 (0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarMeshComponent_TA");
}

return uClassPointer;
};

bool WantsTrailCameraAdjustment();
bool WantsBoostCameraAdjustment();
void UpdateWheelPreviewRotation(float DeltaTime, float PreviewFrontWheelSpinSpeed, float PreviewBackWheelSpinSpeed, unsigned long bIgnoreBoostRequirement);
void SetBodyFXActor(class AFXActor_X* InBodyFX);
void ApplyPaintSettings();
void InitBodyVisuals();
void InitAttachments(class UProductLoader_TA* InLoadout);
void SetLoadout(class UProductLoader_TA* InLoadout);
void SetProductAttributes(struct FClientLoadoutOnlineData& OnlineLoadout);
void InitWheelControl(int32_t Idx, class FString Prefix);
void InitWheelControls();
void InitBoostControls();
void ClearAttachments();
void NotifyWhenAttached(struct FScriptDelegate Callback);
void eventOnDetached();
void eventOnAttached();
void EventAttached(class UCarMeshComponent_TA* Mesh);
};

```

```

// Class TAGame.CarPreviewMeshComponent_TA
// 0x0020 (0x08D0 - 0x08F0)
class UCarPreviewMeshComponent_TA : public UCarMeshComponentBase_TA
{
public:
    class UProductAsset_Boost_TA*           BoostAsset;          // 0x08D0 (0x0008)
    [0x0000000000000000]
    class AFXActor_Boost_TA*               BoostFX;           // 0x08D8 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class UProductAsset_Attachment_TA*    AntennaAsset;       // 0x08E0
    (0x0008) [0x0000000000000000]
    class UProductAsset_Attachment_TA*    HatAsset;           // 0x08E8
    (0x0008) [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CarPreviewMeshComponent_TA");
        }

        return uClassPointer;
    };

    void CreateWheelMeshes();
    void eventInitAttachments();
    void InitFXStatesForActor(class AFXActor_X* FXActor);
    void InitFXStates();
    void InitBoostFX();
    void InitBodyFX();
    void InitFXVisuals();
};

// Class TAGame.SkeletalMeshComponent_SplitBody_TA
// 0x0038 (0x07B8 - 0x07F0)
class USkeletalMeshComponent_SplitBody_TA : public USkeletalMeshComponent_TA
{
public:
    struct FName                   CurrentAnim;          // 0x07B8 (0x0008)
    [0x0000000000000000]
    struct FName                   AnimToSyncWith;      // 0x07C0 (0x0008)
    [0x0000000000000000]
    struct FName                   SplitAttachBone;    // 0x07C8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class FString                 SplitPrimarySuffix; // 0x07D0 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString                 SplitSecondarySuffix; // 0x07E0 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkeletalMeshComponent_SplitBody_TA");
}

return uClassPointer;
};

void SyncPlayAnim(struct FName AnimName, struct FName SyncAnimName, unsigned long
bLoop);
void InitVisuals();
void InitFromAttached();
};

// Class TAGame.SkyLightVolumeComponent_TA
// 0x0060 (0x01D0 - 0x0230)
class USkyLightVolumeComponent_TA : public USkyLightComponent
{
public:
float BlendDistance; // 0x01D0 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class ASkyLight* LightToOverride; // 0x01D8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UBrushComponent* BrushComponent; // 0x01E0
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
uint8_t UnknownData00[0x8]; // 0x01E8 (0x0008) MISSED
OFFSET
struct FMatrix CachedParentToWorld; // 0x01F0 (0x0040)
[0x000000000002002] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkyLightVolumeComponent_TA");
}

return uClassPointer;
};

void SetLowerLightProperties(float NewBrightness, struct FColor NewLightColor);
};

// Class TAGame.SpecialEditionDatabase_TA
// 0x0064 (0x0060 - 0x00C4)
class USpecialEditionDatabase_TA : public UObject
{
public:

```

```

TArray<class UProductSpecialEdition_TA*> SpecialEditions; // 0x0060
(0x0010) [0x00000000000420003] (CPF>Edit | CPF_Const | CPF>EditConst | CPF_NeedCtorLink)
struct FMap_Mirror EditionNameToEditionID; // 0x0070 (0x0050)
[0x00000000000001002] (CPF_Const | CPF_Native)
unsigned long bClickToRebuild : 1; // 0x00C0 (0x0004)
[0x00000000000000001] [0x000000001] (CPF>Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialEditionDatabase_TA");
}

return uClassPointer;
};

class UProductSpecialEdition_TA* GetSpecialEditionByName(struct FName EditionName);
class UProductSpecialEdition_TA* GetSpecialEdition(int32_t Id);
struct FName GetSpecialEditionName(int32_t EditionID);
int32_t GetSpecialEditionId(struct FName EditionName);
};

// Class TAGame.StatEvent_TA
// 0x0060 (0x0060 - 0x00C0)
class UStatEvent_TA : public UObject
{
public:
int32_t Points; // 0x0060 (0x0004)
[0x0000000000000003] (CPF>Edit | CPF_Const)
float CooldownSeconds; // 0x0064 (0x0004)
[0x0000000000000003] (CPF>Edit | CPF_Const)
unsigned long bAddToScore : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000001] (CPF>Edit | CPF_Const)
unsigned long bIsLeaderboardStat : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000002] (CPF>Edit | CPF_Const)
unsigned long bNotifyTicker : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000004] (CPF>Edit | CPF_Const)
unsigned long bShowOnHUD : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000008] (CPF>Edit | CPF_Const)
unsigned long bPrimaryStat : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000010] (CPF>Edit | CPF_Const)
unsigned long bSkipReplication : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000020] (CPF>Edit | CPF_Const)
unsigned long bCanMute : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000040] (CPF>Edit | CPF_Const)
unsigned long bCountMultiplied : 1; // 0x0068 (0x0004)
[0x0000000000000003] [0x00000080] (CPF>Edit | CPF_Const)
class UStatGroup_TA* Group; // 0x0070 (0x0008)
class UTexture* Texture; // 0x0078 (0x0008)

```

```
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FName                      UISound;           // 0x0080 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class UClass*                     DisplayProductStatClass; // 0x0088 (0x0008)
[0x0000000000000003] (CPF_Edit | CPF_Const)
class FString                     Label;             // 0x0090 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                     PluralLabel;        // 0x00A0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                     Description;       // 0x00B0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatEvent_TA");
}

return uClassPointer;
};

float GetNextCooldownTime();
int32_t GetPoints(class AGameEvent_TA* GameEvent, int32_t Count);
struct FName GetGroupName();
};

// Class TAGame.StatFactoryBase_TA
// 0x0010 (0x0268 - 0x0278)
class AStatFactoryBase_TA : public AActor
{
public:
TArray<class UStatEvent_TA*>          AllStatEvents;      // 0x0268 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatFactoryBase_TA");
}

return uClassPointer;
};

};

// Class TAGame.StatGraph_TA
```

```

// 0x0020 (0x0070 - 0x0090)
class UStatGraph_TA : public UComponent
{
public:
class USampleRecordSettings_TA*           RecordSettings;          // 0x0070
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FDouble                          LastTickTime;           // 0x0078 (0x0008)
[0x0000000000000002] (CPF_Const)
TArray<class USampleHistory_TA*>       SampleHistories;        // 0x0080
(0x0010) [0x000000004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatGraph_TA");
}

return uClassPointer;
};

void StopDrawing();
class USampleHistory_TA* CreateSampleHistory(class FString Title);
class USampleHistory_TA* AddSampleHistory(class USampleHistory_TA* History);
void eventConstruct();
};

// Class TAGame.InputBufferGraph_TA
// 0x0028 (0x0090 - 0x00B8)
class UIInputBufferGraph_TA : public UStatGraph_TA
{
public:
class USampleHistory_TA*           Buffer;                  // 0x0090 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*           BufferTarget;           // 0x0098 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*           OverUnderFrames;        // 0x00A0
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*           PhysicsRate;            // 0x00A8 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
float                           MaxPhysicsRate;          // 0x00B0 (0x0004)
[0x0000000000000000]
float                           MinPhysicsRate;          // 0x00B4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InputBufferGraph_TA");
}

return uClassPointer;
};

void AddSample(struct FInputBufferStats& Stats);
class USampleHistory_TA* CreateBufferHistory(class FString Title);
void eventConstruct();
};

// Class TAGame.NetStatGraph_TA
// 0x0044 (0x0090 - 0x00D4)
class UNetStatGraph_TA : public UStatGraph_TA
{
public:
class USampleHistory_TA*          PacketsOut;           // 0x0090 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*          PacketsIn;            // 0x0098 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*          LostPacketsOut;      // 0x00A0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*          LostPacketsIn;       // 0x00A8 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*          BytesOut;             // 0x00B0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*          BytesIn;              // 0x00B8 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA*          Latency;              // 0x00C0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
float                            ExpectedOutPacketRate; // 0x00C8 (0x0004)
[0x0000000000000000]
float                            ExpectedInPacketRate; // 0x00CC (0x0004)
[0x0000000000000000]
float                            MaxBytesRate;         // 0x00D0 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NetStatGraph_TA");
}

return uClassPointer;
};

void eventUpdateGraphRanges();

```

```

class USampleHistory_TA* CreateBytesSummary(class FString Title);
class USampleHistory_TA* CreateLossSummary(class FString Title);
class USampleHistory_TA* CreatePktSummary(class FString Title);
void eventConstruct();
};

// Class TAGame.PerfStatGraph_TA
// 0x0040 (0x0090 - 0x00D0)
class UPerfStatGraph_TA : public UStatGraph_TA
{
public:
class USampleHistory_TA* FPS; // 0x0090 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA* FrameTime; // 0x0098 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA* GameThreadTime; // 0x00A0
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA* RenderThreadTime; // 0x00A8
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class USampleHistory_TA* GPUFrameTime; // 0x00B0 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<class USampleHistory_TA*> FrameTimeHistories; // 0x00B8
(0x0010) [0x000000004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink |
CPF_EditInline)
float MaxFPS; // 0x00C8 (0x0004)
[0x0000000000000000]
float TargetFPS; // 0x00CC (0x0004)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PerfStatGraph_TA");
}

return uClassPointer;
};

void eventUpdateGraphRanges();
class USampleHistory_TA* CreateFrameTimeHistory(class FString Title);
class USampleHistory_TA* CreateFpsHistory(class FString Title);
void eventConstruct();
};

// Class TAGame.StatGraphDrawer_TA
// 0x0084 (0x0060 - 0x00E4)
class UStatGraphDrawer_TA : public UObject
{
public:
class USampleHistory_TA* History; // 0x0060 (0x0008)

```

```

[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
float MaxSampleAge; // 0x0068 (0x0004)
[0x0000000000000000]
float Scale; // 0x006C (0x0004)
[0x0000000000000000]
float PixelsPerSample; // 0x0070 (0x0004)
[0x0000000000000000]
float ValueStartX; // 0x0074 (0x0004)
[0x0000000000000000]
float MaxCharHeight; // 0x0078 (0x0004)
[0x0000000000000000]
float SpaceWidth; // 0x007C (0x0004)
[0x0000000000000000]
float LabelMaxWidth; // 0x0080 (0x0004)
[0x0000000000000000]
float ValueMaxWidth; // 0x0084 (0x0004)
[0x0000000000000000]
float GridCellWidth; // 0x0088 (0x0004)
[0x0000000000000000]
float GridCellHeight; // 0x008C (0x0004)
[0x0000000000000000]
float GraphWidth; // 0x0090 (0x0004)
[0x0000000000000000]
float GraphHeight; // 0x0094 (0x0004)
[0x0000000000000000]
float GraphStartX; // 0x0098 (0x0004)
[0x0000000000000000]
float GraphEndX; // 0x009C (0x0004)
[0x0000000000000000]
float X; // 0x00A0 (0x0004)
[0x0000000000000000]
float Y; // 0x00A4 (0x0004)
class UCanvas* Canvas; // 0x00A8 (0x0008)
[0x0000000000000000]
TArray<struct FSummaryLabel> SummaryLabels; // 0x00B0
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)
TArray<struct FGraphLine> GridLines; // 0x00C0 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
TArray<struct FGraphLine> SampleLines; // 0x00D0 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
int32_t SampleLineIndex; // 0x00E0 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatGraphDrawer_TA");
}

```

```

return uClassPointer;
};

};

// Class TAGame.StatGraphSystem_TA
// 0x0044 (0x0070 - 0x00B4)
class UStatGraphSystem_TA : public UComponent
{
public:
float GraphSampleTime; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
uint8_t GraphLevel; // 0x0074 (0x0001)
[0x0000004000000000]
class UPerfStatGraph_TA* PerfStatGraph; // 0x0078 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
class UNetStatGraph_TA* NetStatGraph; // 0x0080 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
class UIInputBufferGraph_TA* InputBufferGraph; // 0x0088 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
TArray<class UStatGraph_TA*> StatGraphs; // 0x0090 (0x0010)
[0x000000004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF>EditInline)
TArray<class UStatGraph_TA*> VisibleStatGraphs; // 0x00A0
(0x0010) [0x000000004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF>EditInline)
int32_t PreallocGraphLines; // 0x00B0 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatGraphSystem_TA");
}

return uClassPointer;
};

void __StatGraphSystem_TA_SetGraphLevel_0x1(class UStatGraph_TA* G);
void PacketReceived(float Latency);
void GraphTime(float Seconds);
void StatGraphNext();
float GetGraphSampleTime(uint8_t Level);
TArray<class UStatGraph_TA*> GetDisplayGraphs(uint8_t Level);
void SetGraphLevel(uint8_t Level);
void eventConstruct();
};

// Class TAGame.StatGroup_TA
// 0x0000 (0x0060 - 0x0060)

```

```

class UStatGroup_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatGroup_TA");
}

return uClassPointer;
};

};

// Class TAGame.StaticMeshComponent_TA
// 0x0004 (0x0300 - 0x0304)
class UStaticMeshComponent_TA : public UStaticMeshComponent
{
public:
unsigned long           bIgnoreScriptAttach : 1;           // 0x0300 (0x0004)
[0x0000000000002002] [0x00000001] (CPF_Const | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StaticMeshComponent_TA");
}

return uClassPointer;
};

void SetBoundsScale(float Value);
void eventOnDetached();
void eventOnAttached();
};

// Class TAGame.NameplateMeshComponent_TA
// 0x0038 (0x0304 - 0x033C)
class UNameplateMeshComponent_TA : public UStaticMeshComponent_TA
{
public:
struct FInterpCurveFloat      DistanceScaleCurve;          // 0x0308 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat      FOVScaleCurve;              // 0x0320 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

```

```

float PlayerOptionsScale; // 0x0338 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NameplateMeshComponent_TA");
}

return uClassPointer;
};

void TickNameplate();
void SetPlayerIndex(int32_t Index);
void SetRenderTexture(class UTextureRenderTarget2D* InTex);
void HandleNameplateScaleModifierUpdated(class UGameViewportClient_TA* ViewportClient);
void eventOnDetached();
void eventOnAttached();
};

// Class TAGame.TargetIndicator_TA
// 0x002C (0x0304 - 0x0330)
class UTargetIndicator_TA : public UStaticMeshComponent_TA
{
public:
float MinTranslationX; // 0x0308 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float MaxTranslationX; // 0x030C (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float MinDistance; // 0x0310 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float MaxDistance; // 0x0314 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float ZTranslationOffset; // 0x0318 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
float MaxDisplayDistance; // 0x031C (0x0004)
[0x0000000000000002] (CPF_Const)
float DistanceScaleMultiplier; // 0x0320 (0x0004)
[0x0000000000000002] (CPF_Const)
class AActor* TargetActor; // 0x0328 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TargetIndicator_TA");
}

```

```

}

return uClassPointer;
};

void SetTarget(class UTarget_TA* Target);
};

// Class TAGame.Team_TA
// 0x01D8 (0x0290 - 0x0468)
class ATeam_TA : public ATeamInfo
{
public:
class UCarColorSet_TA*
[0x0000000000000001] (CPF_Edit)           CarColorSet;          // 0x0290 (0x0008)
class UCarColorSet_TA*
[0x0000000000000001] (CPF_Edit)           CarColorSet_v1;        // 0x0298 (0x0008)
class UCarColorSet_TA*
[0x0000000000000001] (CPF_Edit)           CarColorSet_v2;        // 0x02A0 (0x0008)
struct FLinearColor                      FontColor;            // 0x02A8 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor                      ColorBlindFontColor; // 0x02B8 (0x0010)
[0x0000000000000001] (CPF_Edit)
class UCrossplayConfig_X*
[0x0000800000000000]                     Crossplay;           // 0x02C8 (0x0008)
struct FColor                           TeamControllerColor; // 0x02D0 (0x0004)
[0x0000004000002000] (CPF_Transient)
struct FColor                           TeamScoreStrobeColor; // 0x02D4 (0x0004)
[0x0000004000002000] (CPF_Transient)
class UTeamColorScriptedTexture_TA*
(0x0008) [0x0000000000000000]             TeamScriptedTexture; // 0x02D8
TArray<struct FLinearColor>              DefaultColorList;    // 0x02E0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FLinearColor>              ColorBlindColorList; // 0x02F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FLinearColor>              CurrentColorList;   // 0x0300 (0x0010)
[0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
class AGameEvent_Team_TA*                 GameEvent;           // 0x0310 (0x0008)
[0x0000008100002020] (CPF_Net | CPF_Transient)
TArray<class APRI_TA*>                  Members;            // 0x0318 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class APRI_TA*>                  FormerMembers;     // 0x0328 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                          CustomTeamName;     // 0x0338 (0x0010)
[0x0000004100402020] (CPF_Net | CPF_Transient | CPF_NeedCtorLink)
class UTeamNameComponent_TA*             TeamNameComponent; // 0x0348 (0x0008) [0x000008004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
uint64_t                               ClubID;             // 0x0350 (0x0008)
[0x0001004100000020] (CPF_Net)
class UProductAsset_Lого_TA*            Logo;               // 0x0358 (0x0008)
[0x0000004000000000]
struct FReplicatedLogoData            LogoData;           // 0x0360 (0x0008)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FClubColorSet                   ClubColors;         // 0x0368 (0x0008)

```

```

[0x0008008100002020] (CPF_Net | CPF_Transient)
class AVoteActor_TA* ForfeitVoteArchetype; // 0x0370 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AVoteActor_TA* ForfeitVote; // 0x0378 (0x0008)
[0x0000004000002000] (CPF_Transient)
class AVoteActor_TA* PartyUpVote; // 0x0380 (0x0008)
[0x0000004000002000] (CPF_Transient)
unsigned long bForfeit : 1; // 0x0388 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
TArray<struct FTemporarySpawnSpot> TemporarySpawnSpots; // 0x0390
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t Difficulty; // 0x03A0 (0x0004)
[0x0008000000002020] (CPF_Net | CPF_Transient)
struct FScriptDelegate __EventMemberAdded__Delegate; // 0x03A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventMemberRemoved__Delegate; // 0x03C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventColorsChanged__Delegate; // 0x03D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventScoreUpdated__Delegate; // 0x03F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventLogoChanged__Delegate; // 0x0408
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventForfeit__Delegate; // 0x0420 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __ClubColors__ChangeNotify; // 0x0438
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __Difficulty__ChangeNotify; // 0x0450 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Team_TA");
}

return uClassPointer;
};

void __Team_TA__PostBeginPlay_0x1(class UObject* _);
bool __Team_TA__UpdateCustomColors_0x1(class ATeam_TA* Team);
struct FUniqueNetId __Team_TA__GetPartyFollowerIds_0x2(class APRI_TA* Follower);
bool __Team_TA__GetHumanPlayers_0x1(class APRI_TA* PRI);
bool __Team_TA__GetHumanPrimaryPlayers_0x1(class APRI_TA* PRI);
bool __Team_TA__GetHumanPartyLeaders_0x1(class APRI_TA* PRI);
bool __Team_TA__GetNumOfMembersThatCanStartForfeit_0x1(class APRI_TA* P);
void __Team_TA__OnPlayerMarkedAsIdleBanned_0x2(class APRI_TA* M);
void __Team_TA__EnableAllMembersStartVoteToForfeit_0x1(class APRI_TA* Member);
void __Difficulty__ChangeNotifyFunc();
void __ClubColors__ChangeNotifyFunc();

```

```
void UpdateGameShaderParamColors(int32_t TeamIdx, TArray<struct FLinearColor> NewColors);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void OnClubColorsChanged();
void SetClubColors(struct FClubColorSet Colors);
void Forfeit();
void HandleForfeitVoteDestroyed(class AVoteActor_TA* VoteActor);
void EnableAllMembersStartVoteToForfeit();
void OnPlayerMarkedAsIdleBanned(class APRI_TA* IdlePlayer);
void EnableAllMembersStartVoteToForfeitIfNecessary();
void HandleForfeitVoteFinished(class AVoteActor_TA* VoteActor);
int32_t GetNumOfMembersThatCanStartForfeit();
void VoteToForfeit(class APRI_TA* PRI);
void GetColors(struct FLinearColor& OutFontColor, TArray<struct FLinearColor>& OutColors);
void NotifyKismetTeamColorChanged();
void UpdateColors();
void SetLogo(int32_t Logoid, unsigned long bSwapColors);
class FString GetTeamName();
void SetClubID(uint64_t InClubID);
void SetCustomTeamName(class FString NewName);
void SetDefaultColors();
void SetColorList(TArray<struct FLinearColor> ColorList, unsigned long bAllowMaxBrightness);
bool IsSingleParty();
TArray<class APRI_TA*> GetHumanPartyLeaders();
TArray<class APRI_TA*> GetHumanPrimaryPlayers();
TArray<class APRI_TA*> GetHumanPlayers();
TArray<struct FUniqueNetId> GetPartyFollowerIds(class APRI_TA* PRI);
class APRI_TA* GetTeamMemberNamed(class FString PlayerName);
int32_t GetNumBots();
int32_t GetNumHumans();
void OnScoreUpdated();
void ResetScore();
void RemovePoints(int32_t Points);
void SetScore(int32_t Points);
void ScorePoint(int32_t AdditionalScore);
void eventDestroyed();
void MuteOtherTeam(class ATeam_TA* OtherTeam, unsigned long bMute);
void MutePlayer(class AController* Other);
bool GetPlayerReplacementSpot(struct FVector& OutLoc, struct FRotator& OutRot, struct FVector& OutVel);
void ClearTemporarySpawnSpots();
void ExpireTemporarySpawnSpots();
void AddTemporarySpawnSpot(class AActor* AtActor);
void RemoveFromFormerMembers(class AController* RemovedPlayer);
void RemoveFromTeam(class AController* Other);
bool AddToTeam(class AController* Other);
void OnGameEventSet();
void SetGameEvent(class AGameEvent_Team_TA* InGameEvent);
void UpdateCustomColors();
void InitCustomColorUpdates();
void HandleGameDataSelected(int32_t PlaylistId, int32_t MutatorIndex);
void HandleGRISpawned(class AGRI_X* GRI);
void eventPostBeginPlay();
void eventConstruct();
void eventReplicatedEvent(struct FName VarName);
```

```

void EventForfeit(class ATeam_TA* Team);
void EventLogoChanged(class ATeam_TA* Team);
void EventScoreUpdated(class ATeam_TA* Team);
void EventColorsChanged(class ATeam_TA* Team);
void EventMemberRemoved(class ATeam_TA* Team, class APRI_TA* Member);
void EventMemberAdded(class ATeam_TA* Team, class APRI_TA* Member);
};

// Class TAGame.TeamColorScriptedTexture_TA
// 0x0014 (0x01B8 - 0x01CC)
class UTeamColorScriptedTexture_TA : public UScriptedTexture
{
public:
TArray<struct FLinearColor> PixelColorList; // 0x01B8 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bMaxBrightness : 1; // 0x01C8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TeamColorScriptedTexture_TA");
}

return uClassPointer;
};

void RenderColorArray(TArray<struct FLinearColor> ColorArray);
void OnRender(class UCanvas* Canvas);
static struct FLinearColor GetFullBrightColor(struct FLinearColor C, float ValueScale);
void SetColorsArray(TArray<struct FLinearColor> ColorList, unsigned long bAllowMaxBrightness);
};

// Class TAGame.ThrottleShakeComponent_TA
// 0x000C (0x01B8 - 0x01C4)
class UThrottleShakeComponent_TA : public UShakeComponent_X
{
public:
float ThrottleRiseSpeed; // 0x01B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ThrottleFallSpeed; // 0x01BC (0x0004)
[0x0000000000000001] (CPF_Edit)
float Throttle; // 0x01C0 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ThrottleShakeComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.ThumbnailRenderer_TA
// 0x01D8 (0x0060 - 0x0238)
class UThumbnailRenderer_TA : public UObject
{
public:
struct FRotator           CameraRotation;           // 0x0060 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector            CameraTranslation;        // 0x006C (0x000C)
[0x0000000000000001] (CPF_Edit)
float                      CameraFOV;              // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long              bAutoZoomCamera : 1;    // 0x007C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long              bDisableLightEnvironment : 1; // 0x007C (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long              CompressionNone : 1;     // 0x007C (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long              bSkipBackground : 1;      // 0x007C (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
unsigned long              bSkipScene : 1;          // 0x007C (0x0004)
[0x0000000000000001] [0x00000010] (CPF_Edit)
struct FRotator           ProductRotation;         // 0x0080 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector            ProductTranslation;       // 0x008C (0x000C)
[0x0000000000000001] (CPF_Edit)
class UPrefab*             ScenePrefab;            // 0x0098 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UPostProcessChain*   ScenePostProcessChain;   // 0x00A0
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FPostProcessSettings ScenePostProcessSettings; // 0x00A8
(0x0168) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FLinearColor         BackgroundColor;        // 0x0210 (0x0010)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t                    RenderWidth;            // 0x0220 (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t                    RenderHeight;           // 0x0224 (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t                    TextureWidth;           // 0x0228 (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
int32_t                    TextureHeight;          // 0x022C (0x0004)
[0x0000000000004001] (CPF_Edit | CPF_Config)
uint8_t                     MipGenSettings;        // 0x0230 (0x0001)
[0x0000000000000001] (CPF_Edit)

```

```

int32_t LODBias; // 0x0234 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ThumbnailRenderer_TA");
}

return uClassPointer;
};

class UTexture2D* RenderScene(class UThumbnailScene_TA* Scene, class FString
TextureName, class UPackage* InPackage);
class UThumbnailScene_TA* BuildScene();
};

// Class TAGame.ThumbnailScene_TA
// 0x0048 (0x0060 - 0x00A8)
class UThumbnailScene_TA : public UObject
{
public:
TArray<struct FThumbnailComponent> SceneComponents; // 0x0060
(0x0010) [0x0000000000480000] (CPF_Component | CPF_NeedCtorLink)
struct FVector ViewTranslation; // 0x0070 (0x000C)
[0x0000000000000000]
struct FRotator ViewRotation; // 0x007C (0x000C)
[0x0000000000000000]
unsigned long bForceAllUsedMipsResident : 1; // 0x0088 (0x0004)
[0x0000000000000000] [0x00000001]
class UTexture2D* Background; // 0x0090 (0x0008)
[0x0000000000000000]
struct FPointer RenderScene; // 0x0098 (0x0008)
[0x0000000000003002] (CPF_Const | CPF_Native | CPF_Transient)
float CameraFOV; // 0x00A0 (0x0004)
[0x0000000000000000]
int32_t TeamID; // 0x00A4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ThumbnailScene_TA");
}

return uClassPointer;
};

```

```

};

void HideAllMeshes();
class UActorComponent* GetSceneComponent(class UClass* ComponentClass, struct FName Tag);
};

// Class TAGame.TickNotifyDelegate_TA
// 0x0020 (0x0060 - 0x0080)
class UTickNotifyDelegate_TA : public UObject
{
public:
struct FPointer           VfTable_IITickNotify_TA;          // 0x0060 (0x0008)
[0x0000000000080100] (CPF_Const | CPF_Native | CPF_NoExport)
struct FScriptDelegate     _EventPrePhysicsStep__Delegate;    // 0x0068
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TickNotifyDelegate_TA");
}

return uClassPointer;
};

void Stop();
void Start();
class UTickNotifyDelegate_TA* OnPrePhysicsStep(struct FScriptDelegate Callback);
static class UTickNotifyDelegate_TA* Create();
void EventPrePhysicsStep(float DeltaTime);
};

// Class TAGame.UserBugReportBulkData_TA
// 0x0038 (0x0060 - 0x0098)
class UUserBugReportBulkData_TA : public UObject
{
public:
TArray<struct FUserReportVideoSetting>      VideoSettings;          // 0x0060
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<uint8_t>                          Data;                  // 0x0070 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     _EventComplete__Delegate;        // 0x0080
(0x0018) [0x000000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportBulkData_TA");
}

return uClassPointer;
};

void GatherLog(TArray<uint8_t>& OutData);
void GatherMemoryDump(int32_t MaxPropertyStringLength, TArray<class UClass*>& IgnoreClasses, TArray<uint8_t>& OutData);
void CompressDump(TArray<uint8_t>& UncompressedData, TArray<uint8_t>& CompressedData);
void GatherDump();
void GatherVideoSettings();
void GatherData();
void WaitForGC();
void Init(struct FScriptDelegate CompleteCallback);
void EventComplete(class UUserBugReportBulkData_TA* BulkData);
};

// Class TAGame.VanityQuery_TA
// 0x0058 (0x0070 - 0x00C8)
class UVanityQuery_TA : public UComponent
{
public:
int32_t MaxNumberQueriesAllowed; // 0x0070 (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<struct FVanityQuerySet> ActiveQueries; // 0x0078 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FVanityQueryRequest> QueuedRequests; // 0x0088
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __VanitiesUpdated__Delegate; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __CompleteRequest__Delegate; // 0x00B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VanityQuery_TA");
}

return uClassPointer;
};

void CompleteRequests(TArray<struct FUniqueNetId> PlayerIds, uint8_t InType, struct FScriptDelegate ProcessQuery, unsigned long bUsedCachedResult);
bool GetQueryInfo(struct FUniqueNetId PlayerID, uint8_t InType, class UGFxShell_X*& OutShell, struct FScriptDelegate& OutCallback);
void FindNewRequests(struct FVanityQueryRequest& InRequest, struct FVanityQuerySet&

```

```

QuerySet);
int32_t GetNumActiveQueries();
void GetNextAvailableIDs(TArray<struct FVanityQuerySet>& QuerySets);
bool CanStartNewQuery();
void StartQuery(class UGFxShell_X* Shell, TArray<struct FUniqueNetId> PlayerIds, uint8_t InType,
struct FScriptDelegate Callback, unsigned long bForcedUpdate);
bool HaveRequestsQueued();
int32_t FindQueryIndex(struct FUniqueNetId InPlayerId, uint8_t InVanityType);
void CompleteRequest(TArray<struct FUniqueNetId> PlayerIds, class UGFxShell_X* Shell, struct
FScriptDelegate Callback);
void VanitiesUpdated(TArray<struct FUniqueNetId> PlayerIds);
};

// Class TAGame.VanitySet_TA
// 0x0018 (0x0060 - 0x0078)
class UVanitySet_TA : public UObject
{
public:
class UPlayerAvatar_TA*           Avatar;          // 0x0060 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPlayerBanner_TA*           Banner;         // 0x0068 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPlayerAvatarBorder_TA*     Border;         // 0x0070 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VanitySet_TA");
}

return uClassPointer;
};

};

// Class TAGame.VanitySetManager_TA
// 0x0078 (0x0060 - 0x00D8)
class UVanitySetManager_TA : public UObject
{
public:
class UOnlineGame_X*              OnlineGame;      // 0x0060 (0x0008)
[0x0000800000000000]
struct FMap_Mirror                VanityMap;       // 0x0068 (0x0050)
[0x0000000000001000] (CPF_Native)
class UVanityQuery_TA*             VanityQuery;    // 0x00B8 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FScriptDelegate              _VanitiesUpdated__Delegate; // 0x00C0
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VanitySetManager_TA");
}

return uClassPointer;
};

bool __VanitySetManager_TA__ReadVanity_Internal_0x1(struct FUniqueNetId P);
bool __VanitySetManager_TA__HandlePsyNetReadBanner_0x1(int32_t Type);
bool __VanitySetManager_TA__HandlePsyNetReadBorder_0x1(int32_t Type);
void SetBorder(class UPlayerAvatarBorder_TA* InBorder);
void HandleLoadedAvatarBorderAsset(class UPlayerVanity_TA* PlayerVanity);
void HandlePsyNetReadBorder(class URPC_PsyNetGetVanities_TA* RPC);
void RequestsBorders(TArray<struct FUniqueNetId> PlayerIds, unsigned long bForcedUpdate,
class URPC_PsyNetGetVanities_TA*& RPC);
void ProcessBorders(TArray<struct FUniqueNetId> PlayerIds, class UGFxShell_X* Shell, struct
FScriptDelegate Callback);
void ReadBorders(class UGFxShell_X* Shell, TArray<struct FUniqueNetId> PlayerIds, struct
FScriptDelegate Callback, unsigned long bForceUpdate);
class UPlayerAvatarBorder_TA* GetAvatarBorder(struct FUniqueNetId PlayerID);
void SetBanner(class UPlayerBanner_TA* InBanner);
void HandleLoadedBannerAsset(class UPlayerVanity_TA* PlayerVanity);
void HandlePsyNetReadBanner(class URPC_PsyNetGetVanities_TA* RPC);
void RequestBanners(TArray<struct FUniqueNetId> PlayerIds, unsigned long bForcedUpdate,
class URPC_PsyNetGetVanities_TA*& RPC);
void ProcessBanners(TArray<struct FUniqueNetId> PlayerIds, class UGFxShell_X* Shell, struct
FScriptDelegate Callback);
void ReadBanners(class UGFxShell_X* Shell, TArray<struct FUniqueNetId> PlayerIds, struct
FScriptDelegate Callback, unsigned long bForcedUpdate);
class UPlayerBanner_TA* GetBanner(struct FUniqueNetId PlayerID);
void SetAvatar(class UPlayerAvatar_TA* InAvatar);
void HandleLoadedAvatarAsset(class UPlayerVanity_TA* PlayerVanity);
void HandleSubSystemReadAvatar(struct FUniqueNetId PlayerID, class UTexture* Avatar, class
FString OnlinePlayerName);
void RequestAvatars(TArray<struct FUniqueNetId> PlayerIds, unsigned long bForcedUpdate,
class URPC_PsyNetGetVanities_TA*& RPC);
void ProcessAvatars(TArray<struct FUniqueNetId> PlayerIds, class UGFxShell_X* Shell, struct
FScriptDelegate Callback);
void ReadAvatars(class UGFxShell_X* Shell, TArray<struct FUniqueNetId> PlayerIds, struct
FScriptDelegate Callback, unsigned long bForceUpdate);
class UPlayerAvatar_TA* GetAvatar(struct FUniqueNetId PlayerID);
void HandlePsyNetLoggedIn();
void ReadVanity_Internal(class UGFxShell_X* Shell, TArray<struct FUniqueNetId> PlayerIds,
uint8_t InType, struct FScriptDelegate Callback, unsigned long bForcedUpdate);
void TryProcessVanity(uint8_t InType, TArray<struct FUniqueNetId>& PlayerIds);
void UpdateQueries(float DeltaTime);
class UVanitySet_TA* GetOrCreateVanity(struct FUniqueNetId PlayerID);
void AddToMap(struct FUniqueNetId PlayerID, class UVanitySet_TA* Vanity);

```

```

class UVanitySet_TA* GetVanitySet(struct FUniqueNetId PlayerID);
void VanitiesUpdated(TArray<struct FUniqueNetId> PlayerIDs);
};

// Class TAGame.VehiclePartStaticMeshComponent_TA
// 0x0008 (0x0300 - 0x0308)
class UVehiclePartStaticMeshComponent_TA : public UStaticMeshComponent
{
public:
class URB_BodySetup*           BodySetupOverride;          // 0x0300 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VehiclePartStaticMeshComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.VehiclePickup_TA
// 0x0088 (0x0268 - 0x02F0)
class AVehiclePickup_TA : public AActor
{
public:
float                      RespawnDelay;                // 0x0268 (0x0004)
[0x0000000000000001] (CPF_Edit)

class AFXActor_X*            FXActorArchetype;        // 0x0270 (0x0008)
[0x0000000000000001] (CPF_Edit)

class AFXActor_X*            FXActor;                  // 0x0278 (0x0008)
[0x000000000220001] (CPF_Edit | CPF_EditConst)

class UFXActorEvent_X*       PickedUpEvent;           // 0x0280 (0x0008)
[0x0000000000000000]

class UFXActorEvent_X*       LocalPickedUpEvent;      // 0x0288 (0x0008)
[0x0000000000000000]

class UCylinderComponent*    CylinderComponent;        // 0x0290
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

struct FPickupData           ReplicatedPickupData;    // 0x0298 (0x0010)
[0x0000004100002020] (CPF_Net | CPF_Transient)

struct FPickupData2          NewReplicatedPickupData; // 0x02A8
(0x0010) [0x0000004100002020] (CPF_Net | CPF_Transient)

unsigned long                 bNetRelevant : 1;        // 0x02B8 (0x0004)
[0x0000004000000000] [0x00000001]

unsigned long                 bNoPickup : 1;          // 0x02B8 (0x0004)
[0x0000004100002020] [0x00000002] (CPF_Net | CPF_Transient)

struct FScriptDelegate        _EventSpawned__Delegate; // 0x02C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

struct FScriptDelegate           __EventPickedUp__Delegate;           // 0x02D8
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VehiclePickup_TA");
}

return uClassPointer;
};

void PlayPickedUpFX();
bool IsTouchingAVehicle();
void UpdateTickDisabled();
bool IsPickedUp();
void PrintDebugInfo(class UDebugDrawer* Drawer);
void SetNetRelevant(unsigned long bRelevant);
void Respawn();
void UpdatePickupState();
void SetPickedUpDeprecated(unsigned long bNewPickedUp, class ACar_TA* InInstigator);
void Pickup(class ACar_TA* Car);
bool CanPickup(class ACar_TA* Car);
void OnTouch(class ACar_TA* Car);
void eventTouch(class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector HitLocation, struct FVector HitNormal);
void OnPickUp();
void OnSpawn();
void SetNoPickup();
void SetupReplicateNoPickup();
void InitFX();
void eventPostBeginPlay();
void eventPreBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventPickedUp(class AVehiclePickup_TA* Pickup);
void EventSpawned(class AVehiclePickup_TA* Pickup);
};

// Class TAGame.VehiclePickup_Boost_TA
// 0x0011 (0x02F0 - 0x0301)
class AVehiclePickup_Boost_TA : public AVehiclePickup_TA
{
public:
float                      BoostAmount;                         // 0x02F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*          LocalPickupSound;                // 0x02F8 (0x0008)
[0x0000000000000001] (CPF_Edit)
uint8_t                     BoostType;                      // 0x0300 (0x0001)
[0x0000000000000001] (CPF_Edit)
};

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VehiclePickup_Boost_TA");
}

return uClassPointer;
};

void PlayPickedUpFX();
void Pickup(class ACar_TA* Car);
bool CanPickup(class ACar_TA* Car);
};

// Class TAGame.VehicleSim_TA
// 0x00C7 (0x009D - 0x0164)
class UVehicleSim_TA : public UActorComponent
{
public:
TArray<class UWheel_TA*> Wheels; // 0x00A0 (0x0010)
[0x00000000448000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
struct FInterpCurveFloat SteerAngleCurve; // 0x00B0 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float DriveTorque; // 0x00C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat DriveTorqueCurve; // 0x00D0 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float BrakeTorque; // 0x00E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat HandbrakeSteerAngleCurve; // 0x00F0
(0x0018) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInputRate HandbrakeRate; // 0x0108 (0x0008)
[0x0000000000000001] (CPF_Edit)
float StopThreshold; // 0x0110 (0x0004)
[0x0000000000000001] (CPF_Edit)
float IdleBrakeFactor; // 0x0114 (0x0004)
[0x0000000000000001] (CPF_Edit)
float OppositeBrakeFactor; // 0x0118 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bUseAckermannSteering : 1; // 0x011C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bWasAttached : 1; // 0x011C (0x0004)
[0x0000000000002002] [0x00000002] (CPF_Const | CPF_Transient)
float OutputThrottle; // 0x0120 (0x0004)
[0x0000000000002000] (CPF_Transient)
float OutputSteer; // 0x0124 (0x0004)
[0x0000000000002000] (CPF_Transient)
float OutputBrake; // 0x0128 (0x0004)
[0x0000000000002000] (CPF_Transient)

```

```

float OutputHandbrake; // 0x012C (0x0004)
[0x0000000000002000] (CPF_Transient)
class AVehicle_TA* Vehicle; // 0x0130 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ACar_TA* Car; // 0x0138 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FPointer BulletVehicle; // 0x0140 (0x0008)
[0x0000000000003002] (CPF_Const | CPF_Native | CPF_Transient)
float SteeringSensitivity; // 0x0148 (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<float> TireFrictionScales; // 0x0150 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
float TireFrictionScale; // 0x0160 (0x0004)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VehicleSim_TA");
}

return uClassPointer;
};

void RemoveTireFrictionScale(float InScale);
void AddTireFrictionScale(float InScale);
void PrintDebugInfo(class UDebugDrawer* Drawer);
struct FTurningCircle GetMaxTurningCircle(float SteerAmt, float AtSpeed, struct FVector AtLocation, struct FRotator AtRotation);
struct FTurningCircle GetMaxTurningCircleAt(float SteerAmt, float AtSpeed, struct FVector AtLocation, struct FRotator AtRotation);
void eventOnDetached();
void SetupWheelPreset(class UWheel_TA* Wheel, struct FVehicleAxeData PresetAxe, struct FVehicleWheelSettings WheelSettings);
void SetupPresetAxe(class UWheel_TA* LeftWheel, class UWheel_TA* RightWheel, struct FVehicleAxeSettings VisualAxe, struct FVehicleAxeData PresetAxe);
void SetupWheelBone(class UWheel_TA* Wheel, struct FVehicleAxeSettings Axe, struct FVehicleWheelSettings WheelSettings);
void SetupBoneAxe(class UWheel_TA* LeftWheel, class UWheel_TA* RightWheel, struct FVehicleAxeSettings Axe);
void eventOnAttached();
};

// Class TAGame.VoteActor_TA
// 0x0118 (0x0268 - 0x0380)
class AVoteActor_TA : public AActor
{
public:
uint8_t Subject; // 0x0268 (0x0001)
[0x0000004000000001] (CPF_Edit)

```

```

int32_t TimeRemaining; // 0x026C (0x0004)
[0x0000004000000001] (CPF_Edit)
unsigned long bUnanimousVote : 1; // 0x0270 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bAllowSplitscreenVoters : 1; // 0x0270 (0x0004)
[0x0000004000000001] [0x00000002] (CPF_Edit)
unsigned long bFilterIdleBannedPlayers : 1; // 0x0270 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long bFinished : 1; // 0x0270 (0x0004)
[0x0000004100002020] [0x00000008] (CPF_Net | CPF_Transient)
TArray<struct FVoter> Voters; // 0x0278 (0x0010)
[0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FVoter ReplicatedVoters[0x8]; // 0x0288 (0x0080)
[0x0000000100002020] (CPF_Net | CPF_Transient)
struct FScriptDelegate __EventStarted__Delegate; // 0x0308 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventVotersChanged__Delegate; // 0x0320
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTimeRemainingChanged__Delegate; // 0x0338
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventFinished__Delegate; // 0x0350
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventDestroyed__Delegate; // 0x0368
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VoteActor_TA");
}

return uClassPointer;
};

void __VoteActor_TA__OnVotersChanged_0x2(struct FVoter V);
void __VoteActor_TA__OnVotersChanged_0x1(struct FVoter V);
bool __VoteActor_TA__RequiredVotes_0x2(struct FVoter V);
bool __VoteActor_TA__RequiredVotes_0x1(struct FVoter V);
void __VoteActor_TA__BeginState_0x1(struct FVoter V);
void eventDestroyed();
void DestroySelf();
bool Failed();
bool Passed();
int32_t RequiredVotes();
int32_t GetVotes(uint8_t VoteStatus, unsigned long bExcludidleBannedPlayers);
void FinishVote();
void CheckFinished();
void SetVoteStatus(class APRI_TA* PRI, uint8_t Status);
void OnVotersChanged();
void OnIdlePlayer(class APRI_TA* PRI);

```

```

void RemoveVoter(class APRI_TA* PRI);
void AddVoter(class APRI_TA* PRI);
void UpdateTimeRemaining();
void AddTeam(class ATeam_TA* Team);
void AddGameEvent(class AGameEvent_TA* GameEvent);
void eventReplicatedEvent(struct FName VarName);
void eventPostBeginPlay();
void EventDestroyed(class AVoteActor_TA* VoteActor);
void EventFinished(class AVoteActor_TA* VoteActor);
void EventTimeRemainingChanged(class AVoteActor_TA* VoteActor);
void EventVotersChanged(class AVoteActor_TA* VoteActor);
void EventStarted(class AVoteActor_TA* VoteActor);
};

// Class TAGame.WebBrowser_TA
// 0x0050 (0x0060 - 0x00B0)
class UWebBrowser_TA : public UObject
{
public:
    unsigned long           bBrowserLoading : 1;           // 0x0060 (0x0004)
    [0x0000004000000000] [0x00000001]
    struct FScriptDelegate      __EventWebBrowserOpened__Delegate;   // 0x0068
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate      __EventWebBrowserClosed__Delegate;   // 0x0080
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate      __EventWebBrowserLoaded__Delegate;   // 0x0098
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.WebBrowser_TA");
        }

        return uClassPointer;
    }

    static void PageLoaded(class FString PageTitle, class FString URL, uint64_t
LoadTimeInSeconds);
    static void Close();
    static void OpenURL(class FString URL, int32_t BrowserWidth, int32_t BrowserHeight);
    void EventWebBrowserLoaded(class FString PageTitle, class FString URL, uint64_t
LoadTimeInSeconds);
    void EventWebBrowserClosed();
    void EventWebBrowserOpened(class FString URL);
};

// Class TAGame.Wheel_TA
// 0x0170 (0x0070 - 0x01E0)
class UWheel_TA : public UComponent

```

```

{
public:
float SteerFactor; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float WheelRadius; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SuspensionStiffness; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SuspensionDampingCompression; // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
float SuspensionDampingRelaxation; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SuspensionTravel; // 0x0084 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SuspensionMaxRaise; // 0x0088 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ContactForceDistance; // 0x008C (0x0004)
[0x0000000000000001] (CPF_Edit)
float PushForceConstant; // 0x0090 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SpinSpeedDecayRate; // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat LatFrictionCurve; // 0x0098 (0x0018)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat LongFrictionCurve; // 0x00B0 (0x0018)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat HandbrakeLatFrictionCurve; // 0x00C8
(0x0018) [0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat HandbrakeLongFrictionCurve; // 0x00E0
(0x0018) [0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat WallFrictionCurve; // 0x00F8 (0x0018)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FName BoneName; // 0x0110 (0x0008)
[0x0000000000000001] (CPF>Edit)
struct FName PhysicsBoneName; // 0x0118 (0x0008)
[0x0000000000000001] (CPF>Edit)
struct FVector BoneOffset; // 0x0120 (0x000C)
[0x0000000000000001] (CPF>Edit)
struct FVector PresetRestPosition; // 0x012C (0x000C)
[0x0000000000000001] (CPF>Edit)
struct FVector LocalSuspensionRayStart; // 0x0138 (0x000C)
[0x0000000000002002] (CPF_Const | CPF_Transient)
struct FVector LocalRestPosition; // 0x0144 (0x000C)
[0x0000000000002002] (CPF_Const | CPF_Transient)
class UVehicleSim_TA* VehicleSim; // 0x0150 (0x0008)
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
int32_t WheelIndex; // 0x0158 (0x0004)
[0x0000000000002002] (CPF_Const | CPF_Transient)
struct FWheelContactData Contact; // 0x0160 (0x0050)
[0x00000000000082002] (CPF_Const | CPF_Transient | CPF_Component)
unsigned long bDrawDebug : 1; // 0x01B0 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long bHadContact : 1; // 0x01B0 (0x0004)
[0x0000000000002002] [0x00000002] (CPF_Const | CPF_Transient)

```

```

float FrictionCurveInput; // 0x01B4 (0x0004)
[0x0000000000000002002] (CPF_Const | CPF_Transient)
float AerialThrottleToVelocityFactor; // 0x01B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AerialAccelerationFactor; // 0x01BC (0x0004)
[0x0000000000000001] (CPF_Edit)
float SpinSpeed; // 0x01C0 (0x0004)
[0x00000000000000002000] (CPF_Transient)
struct FScriptDelegate __EventContactChanged__Delegate; // 0x01C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Wheel_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
struct FVector GetRefWheelLocation();
float GetSuspensionDistance();
float GetSteer();
struct FVector GetLinearVelocity();
void EventContactChanged(class UWheel_TA* Wheel);
};

// Class TAGame.WheelAssetGenerator_TA
// 0x0010 (0x0070 - 0x0080)
class UWheelAssetGenerator_TA : public UComponent
{
public:
TArray<class UWheelAssetSettings_TA*> Wheels; // 0x0070
(0x0010) [0x00000000044A0009] (CPF>Edit | CPF_ExportObject | CPF>EditConst |
CPF_Component | CPF_NeedCtorLink | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelAssetGenerator_TA");
}

return uClassPointer;
};

```

```

void Generate();
void Copy(class UWheelAssetGenerator_TA* Other);
};

// Class TAGame.WheelAssetGenerator_1Wheel_TA
// 0x0010 (0x0080 - 0x0090)
class UWheelAssetGenerator_1Wheel_TA : public UWheelAssetGenerator_TA
{
public:
    uint8_t           MirrorMode;          // 0x0080 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    uint8_t           ParticleMirrorMode; // 0x0081 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    class UWheelAssetSettings_TA*       FrontRight;        // 0x0088 (0x0008)
    [0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.WheelAssetGenerator_1Wheel_TA");
        }

        return uClassPointer;
    };

    void Generate();
    void Copy(class UWheelAssetGenerator_TA* Other);
};

// Class TAGame.WheelAssetGenerator_4Wheels_TA
// 0x0020 (0x0080 - 0x00A0)
class UWheelAssetGenerator_4Wheels_TA : public UWheelAssetGenerator_TA
{
public:
    class UWheelAssetSettings_TA*       FrontLeft;        // 0x0080 (0x0008)
    [0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)
    class UWheelAssetSettings_TA*       FrontRight;       // 0x0088 (0x0008)
    [0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)
    class UWheelAssetSettings_TA*       BackLeft;         // 0x0090 (0x0008)
    [0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)
    class UWheelAssetSettings_TA*       BackRight;        // 0x0098 (0x0008)
    [0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
}

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelAssetGenerator_4Wheels_TA");
}

return uClassPointer;
};

void Generate();
void Copy(class UWheelAssetGenerator_TA* Other);
};

// Class TAGame.WheelAssetGenerator_FrontWheels_TA
// 0x0010 (0x0080 - 0x0090)
class UWheelAssetGenerator_FrontWheels_TA : public UWheelAssetGenerator_TA
{
public:
class UWheelAssetSettings_TA*           FrontRight;          // 0x0080 (0x0008)
[0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)
class UWheelAssetSettings_TA*           FrontLeft;           // 0x0088 (0x0008)
[0x0000000040A0009] (CPF_Edit | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelAssetGenerator_FrontWheels_TA");
}

return uClassPointer;
};

void Generate();
void Copy(class UWheelAssetGenerator_TA* Other);
};

// Class TAGame.WheelAssetGenerator_RightWheels_TA
// 0x0018 (0x0080 - 0x0098)
class UWheelAssetGenerator_RightWheels_TA : public UWheelAssetGenerator_TA
{
public:
uint8_t           MirrorMode;          // 0x0080 (0x0001)
[0x0000000000000001] (CPF_Edit)
uint8_t           ParticleMirrorMode; // 0x0081 (0x0001)
[0x0000000000000001] (CPF_Edit)
class UWheelAssetSettings_TA*           FrontRight;          // 0x0088 (0x0008)

```

```
[0x00000000040A0009] (CPF_Edit | CPF_ExportObject | CPF>EditConst | CPF_Component |  
CPF>EditInline)  
class UWheelAssetSettings_TA*           BackRight;           // 0x0090 (0x0008)  
[0x00000000040A0009] (CPF_Edit | CPF_ExportObject | CPF>EditConst | CPF_Component |  
CPF>EditInline)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.WheelAssetGenerator_RightWheels_TA");  
}  
  
return uClassPointer;  
};  
  
void Generate();  
void Copy(class UWheelAssetGenerator_TA* Other);  
};  
  
// Class TAGame.WheelAssetSettings_TA  
// 0x0050 (0x0070 - 0x00C0)  
class UWheelAssetSettings_TA : public UComponent  
{  
public:  
class UStaticMesh*           StaticMesh;           // 0x0070 (0x0008)  
[0x0000000000000001] (CPF_Edit)           [0x0000000000000001] (CPF_Edit)  
class USkeletalMesh*         SkeletalMesh;         // 0x0078 (0x0008)  
[0x0000000000000001] (CPF_Edit)           [0x0000000000000001] (CPF_Edit)  
class UAnimTree*            AnimTree;            // 0x0080 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
TArray<class UAttachmentBehavior_TA*>   Behaviors;           // 0x0088  
(0x0010) [0x0000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF>EditInline)  
TArray<struct FWheelAttachment>   Attachments;        // 0x0098  
(0x0010) [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)  
struct FVector                 Scale3D;             // 0x00A8 (0x000C)  
[0x0000000000000001] (CPF_Edit)  
struct FRotator                Rotation;           // 0x00B4 (0x000C)  
[0x0000000000000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.WheelAssetSettings_TA");  
}  
  
return uClassPointer;
```

```

};

};

// Class TAGame.WheelSkelMeshMAT_TA
// 0x0000 (0x07B0 - 0x07B0)
class UWheelSkelMeshMAT_TA : public USkeletalMeshComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelSkelMeshMAT_TA");
}

return uClassPointer;
};

};

// Class TAGame.WheelSpeedComponent_TA
// 0x002C (0x00A4 - 0x00D0)
class UWheelspeedComponent_TA : public UActorComponent_X
{
public:
class AVehicle_TA*           Car;           // 0x00A8 (0x0008)
[0x0000000000000000] (CPF_Transient)
float             WheelForwardSpeedInterpRate; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float             WheelSideSpeedInterpRate; // 0x00B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float             MaxWheelSpeed; // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float             AvgWheelForwardSpeed; // 0x00BC (0x0004)
[0x0000000000000000] (CPF_Transient)
float             AvgWheelSideSpeed; // 0x00C0 (0x0004)
[0x0000000000000000] (CPF_Transient)
float             AvgWheelSpinSpeed; // 0x00C4 (0x0004)
[0x0000000000000000] (CPF_Transient)
float             OldWheelRotation; // 0x00C8 (0x0004)
[0x0000000000000000] (CPF_Transient)
float             WheelTurnSpeed; // 0x00CC (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelSpeedComponent_TA");
}

return uClassPointer;
};

void eventDetached();
void eventAttached();
};

// Class TAGame.WheelStaticMeshMAT_TA
// 0x0000 (0x0300 - 0x0300)
class UWheelStaticMeshMAT_TA : public UStaticMeshComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.WheelStaticMeshMAT_TA");
}

return uClassPointer;
};

};

// Class TAGame.BuildInfo_TA
// 0x0038 (0x0060 - 0x0098)
class UBuildInfo_TA : public UObject
{
public:
class FString VersionString; // 0x0060 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
int32_t Changelist; // 0x0070 (0x0004)
[0x0000000000000002] (CPF_Const)
class FString Branch; // 0x0078 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString Label; // 0x0088 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```
uClassPointer = UObject::FindClass("Class TAGame.BuildInfo_TA");
}

return uClassPointer;
};

};

// Class TAGame.RLBot_PreTickActor
// 0x0000 (0x0268 - 0x0268)
class ARLBot_PreTickActor : public AActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RLBot_PreTickActor");
}

return uClassPointer;
};

};

// Class TAGame.RLBot_PostTickActor
// 0x0000 (0x0268 - 0x0268)
class ARLBot_PostTickActor : public AActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RLBot_PostTickActor");
}

return uClassPointer;
};

};

// Class TAGame.DeprecatedProperty
// 0x0000 (0x0070 - 0x0070)
class UDeprecatedProperty : public UField
{
```

```

public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DeprecatedProperty");
}

return uClassPointer;
};

};

// Class TAGame.SkipableProperty
// 0x0018 (0x0070 - 0x0088)
class USkipableProperty : public UDeprecatedProperty
{
public:
uint8_t UnknownData00[0x18]; // 0x0070 (0x0018)
MISSED OFFSET

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkipableProperty");
}

return uClassPointer;
};

};

// Class TAGame.ReplayPackageMap_TA
// 0x01F0 (0x0120 - 0x0310)
class UReplayPackageMap_TA : public UPackageMapLevel
{
public:
uint8_t UnknownData00[0x8]; // 0x0120 (0x0008) MISSED
OFFSET
class UReplay_TA* Replay; // 0x0128 (0x0008)
[0x0000000000000000]
uint8_t UnknownData01[0x20]; // 0x0130 (0x0020)
MISSED OFFSET
TArray<class UObject*> ObjectTable; // 0x0150 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
uint8_t UnknownData02[0x1B0]; // 0x0160 (0x01B0)

```

## MISSED OFFSET

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ReplayPackageMap_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.ConsecutiveMatchTracker_TA  
// 0x0010 (0x0060 - 0x0070)  
class UConsecutiveMatchTracker_TA : public UObject  
{  
public:  
TArray<struct FMatchData> PlayerMatchData; // 0x0060 (0x0010)  
[0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ConsecutiveMatchTracker_TA");  
}  
  
return uClassPointer;  
};  
  
bool __ConsecutiveMatchTracker_TA__HandleGameEnded_0x2(class APRI_TA* PRI);  
void Reset();  
int32_t GetConsecutiveMatchesPlayed(struct FUniqueNetId PlayerID);  
void HandleGameEnded(class AGameEvent_Soccar_TA* GameEvent);  
void eventConstruct();  
};  
  
// Class  
TAGame.__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1_0x1  
// 0x0050 (0x0060 - 0x00B0)  
class  
U__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1_0x1 : public UObject  
{  
public:
```

```

struct FMatchData           Player;          // 0x0060 (0x0050)
[0x00000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1__ConsecutiveMatchTrac
ker_TA__HandleGameEnded_0x1_0x1");
}

return uClassPointer;
};

bool
__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1__ConsecutiveMatchTracker_TA__
HandleGameEnded_0x1_0x1(class APRI_TA* PRI);
};

// Class TAGame.PRI_TA
// 0x0780 (0x0458 - 0x0BD8)
class APRI_TA : public APRI_X
{
public:
int32_t           MatchScore;          // 0x0458 (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
int32_t           MatchGoals;          // 0x045C (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
int32_t           MatchOwnGoals;        // 0x0460 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t           MatchAssists;         // 0x0464 (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
int32_t           MatchSaves;          // 0x0468 (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
int32_t           MatchShots;          // 0x046C (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
int32_t           MatchDemolishes;       // 0x0470 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t           MatchBonusXP;         // 0x0474 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t           MatchBreakoutDamage;   // 0x0478 (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
unsigned long      bMatchMVP : 1;        // 0x047C (0x0004)
[0x0000004100002020] [0x00000001] (CPF_Net | CPF_Transient)
unsigned long      bMatchAdmin : 1;       // 0x047C (0x0004)
[0x0000004100002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long      bLoadoutSet : 1;       // 0x047C (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
unsigned long      bOnlineLoadoutSet : 1;  // 0x047C (0x0004)
[0x0000004100002020] [0x00000008] (CPF_Net | CPF_Transient)

```

```

unsigned long          bLoadoutsSet : 1;           // 0x047C (0x0004)
[0x0000000000002000] [0x00000010] (CPF_Transient)
unsigned long          bOnlineLoadoutsSet : 1;      // 0x047C (0x0004)
[0x0000004000002000] [0x00000020] (CPF_Transient)
unsigned long          bServerSetTeamLoadoutLocked : 1; // 0x047C
(0x0004) [0x0000000000002000] [0x00000040] (CPF_Transient)
unsigned long          bTeamPaintSet : 1;           // 0x047C (0x0004)
[0x0000000000002000] [0x00000080] (CPF_Transient)
unsigned long          bReady : 1;                 // 0x047C (0x0004)
[0x0000004100002020] [0x00000100] (CPF_Net | CPF_Transient)
unsigned long          bIsDistracted : 1;          // 0x047C (0x0004)
[0x000000100002020] [0x00000200] (CPF_Net | CPF_Transient)
unsigned long          bUsingSecondaryCamera : 1; // 0x047C (0x0004)
[0x0000004000002000] [0x00000400] (CPF_Transient)
unsigned long          bUsingBehindView : 1;        // 0x047C (0x0004)
[0x0000004000002000] [0x00000800] (CPF_Transient)
unsigned long          bUsingFreecam : 1;           // 0x047C (0x0004)
[0x0000004000002000] [0x00001000] (CPF_Transient)
unsigned long          bIsInSplitScreen : 1;        // 0x047C (0x0004)
[0x0000004100000020] [0x00002000] (CPF_Net)
unsigned long          bStartVoteToForfeitDisabled : 1; // 0x047C (0x0004)
[0x0000004100002020] [0x00004000] (CPF_Net | CPF_Transient)
unsigned long          bUsingItems : 1;             // 0x047C (0x0004)
[0x0000000000002020] [0x00008000] (CPF_Net | CPF_Transient)
unsigned long          PlayerHistoryValid : 1;       // 0x047C (0x0004)
[0x0000000100000020] [0x00010000] (CPF_Net)
unsigned long          bIdleBanned : 1;             // 0x047C (0x0004)
[0x0000004100002020] [0x00020000] (CPF_Net | CPF_Transient)
unsigned long          bStayAsPartyActive : 1;       // 0x047C (0x0004)
[0x000800000000002000] [0x00040000] (CPF_Transient)
unsigned long          bAbleToStart : 1;            // 0x047C (0x0004)
[0x0000000000002020] [0x00080000] (CPF_Net | CPF_Transient)
unsigned long          bTeamChanged : 1;           // 0x047C (0x0004)
[0x0000000000002000] [0x00100000] (CPF_Transient)
class AGameEvent_TA*   GameEvent;                // 0x0480 (0x0008)
[0x00000048000002000] (CPF_Transient)
class AGameEvent_TA*   ReplicatedGameEvent;       // 0x0488
(0x0008) [0x0000004100002020] (CPF_Net | CPF_Transient)
class ACar_TA*         Car;                     // 0x0490 (0x0008)
[0x0000008000002000] (CPF_Transient)
struct FLoadoutData    FullLoadouts[0x2];        // 0x0498 (0x0080)
[0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FLoadoutAttributesArray FullLoadoutAttributes[0x2]; // 0x0518
(0x0020) [0x0000008000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FClientLoadoutOnlineData FullOnlineLoadout[0x2]; // 0x0538
(0x0020) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FClientLoadoutData    ClientLoadout;        // 0x0558 (0x0010)
[0x0000004100402020] (CPF_Net | CPF_Transient | CPF_NeedCtorLink)
struct FClientLoadoutOnlineData ClientLoadoutOnline; // 0x0568
(0x0010) [0x0000004100402020] (CPF_Net | CPF_Transient | CPF_NeedCtorLink)
struct FClientLoadoutDatas   ClientLoadouts;        // 0x0578 (0x0020)
[0x0000004100402020] (CPF_Net | CPF_Transient | CPF_NeedCtorLink)
struct FClientLoadoutOnlineDatas ClientLoadoutsOnline; // 0x0598
(0x0028) [0x0000008100402020] (CPF_Net | CPF_Transient | CPF_NeedCtorLink)

```

```

TArray<struct FOnlineProductStat>           OnlineProductStats;          // 0x05C0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
float                                         TimeLoadoutWasLocked;       // 0x05D0 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FLoadoutTeamPaint                    TeamPaint;                  // 0x05D4 (0x0010)
[0x0000000000002000] (CPF_Transient)
TArray<struct FTAPPlayerStat>             MatchStats;                // 0x05E8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                                       WaitingStartTime;          // 0x05F8 (0x0004)
[0x0000004000002000] (CPF_Transient)
float                                         TotalGameTimePlayed;       // 0x05FC (0x0004)
[0x0000000000002000] (CPF_Transient)
class ACameraSettingsActor_TA*            PersistentCamera;          // 0x0600
(0x0008) [0x0000004100002020] (CPF_Net | CPF_Transient)
class ACameraSettingsActor_TA*            Camera;                   // 0x0608 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FProfileCameraSettings            CameraSettings;            // 0x0610 (0x001C)
[0x0000004000002000] (CPF_Transient)
uint8_t                                       CameraPitch;              // 0x062C (0x0001)
[0x0000004000002000] (CPF_Transient)
uint8_t                                       CameraYaw;                // 0x062D (0x0001)
[0x0000004000002000] (CPF_Transient)
uint8_t                                       PawnType;                 // 0x062E (0x0001)
[0x0000000100000020] (CPF_Net)
uint8_t                                       PlayerHistoryKey[0x40];    // 0x062F (0x0040)
[0x0000000100000020] (CPF_Net)
uint8_t                                       ReplicatedWorstNetQualityBeyondLatency; // 0x066F
(0x0001) [0x0008000000000020] (CPF_Net)
uint8_t                                       QuitSeverity;             // 0x0670 (0x0001)
[0x0000004100002020] (CPF_Net | CPF_Transient)
TArray<class UProductStat_TA*>           ProductStats;              // 0x0678 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FProductStat>              PendingProductStatValues; // 0x0688
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FUniqueNetId                      PartyLeader;               // 0x0698 (0x0048)
[0x0000004100400020] (CPF_Net | CPF_NeedCtorLink)
struct FName                            Title;                     // 0x06E0 (0x0008)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FSkillTierData                   SkillTier;                 // 0x06E8 (0x0008)
[0x0000004100002020] (CPF_Net | CPF_Transient)
float                                         DodgeInputThreshold;      // 0x06F0 (0x0004)
[0x0000004000000000]
float                                         SteeringSensitivity;     // 0x06F4 (0x0004)
[0x0000000100002020] (CPF_Net | CPF_Transient)
float                                         AirControlSensitivity;   // 0x06F8 (0x0004)
[0x0000004000002000] (CPF_Transient)
TArray<struct FStatCooldown>            StatCooldowns;             // 0x0700
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                                       BoostPickups;             // 0x0710 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t                                       BallTouches;              // 0x0714 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t                                       CarTouches;               // 0x0718 (0x0004)
[0x0000000000002000] (CPF_Transient)

```

```

class APRI_TA*           ReplacingBotPRI;          // 0x0720 (0x0008)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FMemberTitleStat   PrimaryTitle;           // 0x0728 (0x0020)
[0x0000000100000020] (CPF_Net)
struct FMemberTitleStat   SecondaryTitle;         // 0x0748 (0x0020)
[0x0000000100000020] (CPF_Net)
class UProductAsset_GoalExplosion_TA*   PlayerGoalExplosion;    // 0x0768
(0x0008) [0x0000008000002000] (CPF_Transient)
class UPlayerBanner_TA*        PlayerBanner;        // 0x0770 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPlayerAvatar_TA*        PlayerAvatar;        // 0x0778 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UPlayerAvatarBorder_TA*   PlayerBorder;        // 0x0780 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UProductAsset_MusicStingers_TA*   PlayerMusicStinger; // 0x0788
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FName               BotProductName;          // 0x0790 (0x0008)
[0x0000000100002020] (CPF_Net | CPF_Transient)
int32_t                   BotAvatarProductID;       // 0x0798 (0x0004)
[0x0000004100000020] (CPF_Net)
int32_t                   BotBannerProductID;       // 0x079C (0x0004)
[0x0000004100000020] (CPF_Net)
uint64_t                  ClubID;                 // 0x07A0 (0x0008)
[0x0009000000000020] (CPF_Net)
class FString              PublicIP;                // 0x07A8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t                   SpectatorShortcut;        // 0x07B8 (0x0004)
[0x0000004100002020] (CPF_Net | CPF_Transient)
class UCarDistanceTracker_TA*   CarDistanceTracker; // 0x07C0
(0x0008) [0x0000004000000000]
class AStayAsPartyVoter_TA*    StayAsPartyVoter;    // 0x07C8
(0x0008) [0x0000000000000020] (CPF_Net)
class AStayAsPartyVoteYes_TA*   StayAsPartyVoteYes;  // 0x07D0
(0x0008) [0x0000000000000020] (CPF_Net)
class APickupTimer_TA*        PickupTimer;          // 0x07D8 (0x0008)
[0x0000000000000020] (CPF_Net)
int32_t                   TimeTillItem;           // 0x07E0 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t                   MaxTimeTillItem;         // 0x07E4 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UProductsConfig_TA*     ProductsConfig;       // 0x07E8 (0x0008)
[0x0000800000000000]
class FString              CurrentVoiceRoom;      // 0x07F0 (0x0010)
[0x0001004100400020] (CPF_Net | CPF_NeedCtorLink)
struct FServerSetLoadoutParams PendingLoadout;       // 0x0800
(0x0048) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate        __EventGameEventChanged__Delegate; // 0x0848
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate        __EventStatEvent__Delegate;   // 0x0860
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate        __EventRequestedLoadout__Delegate; // 0x0878
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate        __EventSelectedLoadout__Delegate; // 0x0890
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```
struct FScriptDelegate           __EventVanityChanged__Delegate;      // 0x08A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventStatTickerMessage__Delegate; // 0x08C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventInvalidPsyOnixID__Delegate; // 0x08D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventReadyChanged__Delegate;     // 0x08F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCarPreUpdate__Delegate;    // 0x0908
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCarSet__Delegate;          // 0x0920 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSplitScreenStatusChanged__Delegate; // 0x0938
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventDistracted__Delegate;       // 0x0950
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPersistentCameraSet__Delegate; // 0x0968
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCameraChanged__Delegate;    // 0x0980
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPartyLeaderChanged__Delegate; // 0x0998
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventScorePoint__Delegate;       // 0x09B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventWonMVP__Delegate;          // 0x09C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventStatTitlesSet__Delegate;   // 0x09E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPawnTypeChanged__Delegate; // 0x09F8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCommittedProductStats__Delegate; // 0x0A10
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCommittedMatchStats__Delegate; // 0x0A28
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventStartVoteToForfeitDisabledChanged__Delegate; // 0x0A40
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventTitleChanged__Delegate;      // 0x0A58
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSkillTierChanged__Delegate; // 0x0A70
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventServerChangeTeamFailed__Delegate; // 0x0A88
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventScoredGoal__Delegate;        // 0x0AA0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventReplacingBotChanged__Delegate; // 0x0AB8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventMatchAdmin__Delegate;        // 0x0AD0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventServerUnlockedAchievement__Delegate; // 0x0AE8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSpectatorShortcutChanged__Delegate; // 0x0B00
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventOwnerChanged__Delegate;      // 0x0B18
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
```

```

struct FScriptDelegate           __EventQuitSeverityChanged__Delegate;      // 0x0B30
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventIdleBannedChanged__Delegate;        // 0x0B48
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventCurrentVoiceRoomChanged__Delegate; // 0x0B60 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPlayerGoalExplosionChanged__Delegate; // 0x0B78 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bStayAsPartyActive__ChangeNotify;       // 0x0B90
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate
__ReplicatedWorstNetQualityBeyondLatency__ChangeNotify; // 0x0BA8 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __ClubID__ChangeNotify;                  // 0x0BC0 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PRI_TA");
}

return uClassPointer;
};

void __PRI_TA__PostBeginPlay_0x1(class UOnlineClubCache_X* C, class UClubDetails_X* D);
static bool __PRI_TA__InitClientLoadoutAttributes_0x1(class UProductAttribute_TA* A);
bool __PRI_TA__InitFromAttributes_0x1(class UProductAttribute_TA* A);
void __PRI_TA__UpdatePlayerBanner_0x2(class UPlayerVanity_TA* PlayerVanity);
void __PRI_TA__UpdatePlayerBanner_0x1(class UPlayerVanity_TA* PlayerVanity);
void __PRI_TA__UpdatePlayerAvatar_0x1(class UPlayerVanity_TA* PlayerVanity);
void __PRI_TA__UpdatePlayerAvatarBorder_0x1(class UPlayerVanity_TA* PlayerVanity);
struct FOnlineProductAttribute __PRI_TA__GetBannerData_0x1(class UProductAttribute_TA* A);
struct FOnlineProductAttribute __PRI_TA__GetAvatarBorderData_0x1(class
UProductAttribute_TA* A);
void __ClubID__ChangeNotifyFunc();
void __ReplicatedWorstNetQualityBeyondLatency__ChangeNotifyFunc();
void __bStayAsPartyActive__ChangeNotifyFunc();
TArray<struct FScoreboardStat> GetSpectatorStats();
TArray<struct FScoreboardStat> GetScoreboardStats();
bool ShowOnScoreboard();
int32_t GetPing();
bool AllowCarRespawn();
void eventDestroyed();
struct FOnlineProductData GetAvatarBorderData();
struct FOnlineProductData GetBannerData();
void ClientStayAsPartyLeader(struct FMergePartyReplicatedData EventData);
void ServerInviteeConfirmedMerge();
void ClientStayAsPartyFollower(struct FUniqueNetId NewLeader);
void ServerSetCurrentVoiceRoom(class FString RoomName);

```

```
void SetIdleBanned(unsigned long bInIdleBanned);
void OnQuitSeverityChanged();
void SetQuitSeverity(uint8_t SeverityType);
void OnSpectatorShortcutChanged();
void SetSpectatorShortcut(int32_t InShortcut);
void ServerSetPublicIP(class FString IP);
void OnClubsUpdated();
void OnUniquedChanged();
void HandleSubSystemReadAvatar(struct FUniqueNetId InPlayerId, class UTexture* InAvatar,
class FString InOnlinePlayerName);
void UpdatePlayerAvatarBorder();
void UpdatePlayerAvatar();
void UpdatePlayerBanner();
class FString GetBotName();
void SetBotProductName(struct FName InBotProductName);
void ClientUnlockAchievement(int32_t AchievementId, uint8_t AchievementType);
void UpdateCarLocalPlayer();
void OnReplacingBotPRIChanged();
void OnTeamChanged();
void ClearBotReplacement();
void SetBotReplacement(class AAIController_TA* C);
void ReportCheater(class FString Reason);
void PrintDebugInfo(class UDebugDrawer* Drawer);
class FString GetDebugName();
void OnStartVoteToForfeitDisabledChanged();
void SetStartVoteToForfeitDisabled(unsigned long bDisabled);
void ServerVoteToForfeit();
class UOnlineGame_Base_X* GetOnlineGameBase();
class UOnlineGame_X* GetOnlineGame();
void SetUserCarPreferences(float NewDodgeThreshold, float NewSteeringSensitivity, float
NewAirControlSensitivity);
void ServerSetUserCarPreferences(float NewDodgeThreshold, float NewSteeringSensitivity, float
NewAirControlSensitivity);
int32_t ValidateUserInt(class FString Reason, int32_t newValue, int32_t Min, int32_t Max);
float ValidateUserFloat(class FString Reason, float newValue, float Min, float Max);
void OnPawnTypeChanged();
void SetWaitingPlayer(unsigned long B);
void SetPawnType(uint8_t NewPawnType);
bool IsPlayer();
bool IsEditor();
bool IsSpectator();
bool ShouldChooseTeamWhenSpectating();
void Spectate();
void ServerSpectate();
void ClientChangeTeamFailed(int32_t TeamNum);
void ServerChangeTeam(int32_t TeamNum);
bool AllowTeamChange();
void ServerSetVoteStatus(class AVoteActor_TA* VoteActor, uint8_t NewVote);
void SetPodiumTitle(int32_t Index, struct FMemberTitleStat InTitle);
void ReplicatePodiumTitles();
void ClearPodiumTitle();
void OnMatchAdmin();
void OnMVPChange();
void NotifyWonMVP();
```

```
bool IsInvalidName();
bool IsInGameState(struct FName StateName);
bool IsInWarmUpMode();
bool IsWaitingForTeam();
void SetReady(unsigned long bNewReady);
void ServerMatchAdminSetPaused(unsigned long bPause);
void ServerMatchAdminSetScoreAndTime(int32_t NewScoreTeam0, int32_t NewScoreTeam1,
int32_t SecondsRemaining, unsigned long bOverTime, unsigned long bRestartRound);
void SetMatchAdmin(unsigned long bIsMatchAdmin);
void ServerReadyUp();
bool ShouldBroadCastWelcomeMessage(unsigned long bExiting);
void ClientUpdateOnlineProductStats(struct FClientOnlineProductStat ClientStat);
void UpdateOnlineProductStats();
void ClientGiveOnlineCurrencyDrop(struct FCurrency Drop);
void ClientGiveRewardDrop(struct FOnlineXPReward Drop);
void ClientGiveOnlineReward(struct FOnlineProductData Product);
void HandleOnlineCurrencyDrops(TArray<struct FCurrency>& CurrencyDrops);
void HandleRewardDrops(TArray<struct FOnlineXPReward>& RewardDrops);
void HandleOnlineProductDrops(TArray<struct FOnlineProductData>& ProductDrops);
void SetWaitingStartTime();
void ClientNotifyStatTickerMessage(class APRI_TA* Receiver, class APRI_TA* Victim, class
UStatEvent_TA* StatEvent);
void HandleNetStatsPolled(class UClientConnectionTracker_TA* InTracker, struct
FClientConnectionStats InStats);
void OnRep_RepStatTitles();
void ClientScoredGoal(struct FVector BallHitLocation);
void OnScoredGoal(struct FVector BallHitLocation);
void OnRep_SteeringSensitivity();
void OnRep_ClientScorePoint();
void ResetScore();
void RemovePoints(int32_t Points);
void ScorePoint(int32_t AdditionalScore);
void SetStatCount(class UStatEvent_TA* StatEvent, int32_t ForceCount);
void IncrementStat(class UStatEvent_TA* StatEvent, int32_t Count);
void ClientNotifyGainedStat(class UStatEvent_TA* StatEvent, int32_t Count);
void OnGainedStat(class UStatEvent_TA* StatEvent, int32_t Count);
void GiveScore(class UStatEvent_TA* StatEvent, int32_t Count);
bool TryAwardTimeRestrictedStat(class UStatEvent_TA* Stat);
void ResetAllStatCooldowns();
int32_t GetMatchStatCount(struct FName StatName);
void CommitMatchStats();
void CommitProductStats();
void HandleRemoved();
int32_t GetMatchXP();
void InitMusicStingers(class UProductLoader_TA* Loadout);
void InitGoalExplosion(class UProductLoader_TA* Loadout);
void InitProductStats(class UProductLoader_TA* Loadout);
void HandleLoadoutLoaded(class UProductLoader_TA* Loadout);
void UpdateFromLoadout();
void UpdateUserCarPreferences(class ACarComponent_AirControl_TA* AirControlComponent);
void HandleAirControl(class ACarComponent_AirControl_TA* AirControlComp);
void SetCar(class ACar_TA* NewCar);
bool IsClientPlayerPRI();
bool IsLocalPlayerPRI();
```

```
void Unregister();
void AddLocalPlayerToGameEvent();
void SetReplayGameEvent(class AGameEvent_TA* InGameEvent);
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void ValidateReplicatedLoadout(int32_t TeamIndex);
void OnLoadoutsSetInternal();
bool AreLoadoutsSet();
void OnLoadoutsOnlineSet();
void OnLoadoutsSet();
void Deprecated_OnLoadoutSetInternal();
bool Deprecated_IsLoadoutSet();
void Deprecated_OnLoadoutOnlineSet();
void Deprecated_OnLoadoutSet();
void SetTestLoadout(struct FLoadoutData NewLoadout);
TArray<struct FOnlineProductData> GetMatchCompleteLoadout();
int32_t GetLoadoutSlotForCurrentTeam(int32_t InSlotIndex);
void InitCertifiedProductStat(struct FProductInstanceID InstanceID, int32_t ProductID, class UProductAttribute_Certified_TA* Certified);
void InitFromAttributes(struct FLoadoutData& Loadout, struct FLoadoutAttributesArray& LoadoutAttributes);
static void InitClientLoadoutAttributes(struct FLoadoutAttributesArray& LoadoutAttributes, struct FClientLoadoutOnlineData& OutClientData);
void OverrideLoadoutsByPlaylist();
class UGameSettingPlaylist_TA* GetPlaylist();
void OnLoadoutCorrected(int32_t TeamIndex);
void HandleOnlineLoadoutReceived(class URPC_ProductsLoadoutGet_TA* RPC);
void InitLoadoutAttributesForTeam(class APRI_X* PRI);
void HandleDLCPaintModified();
void HandleDLCLoadoutModified();
void ValidateLoadout();
void SetLoadoutSlotToDefaultID(int32_t TeamIndex, class UProductSlot_TA*& Slot);
void SetLoadouts(struct FLoadoutData& Loadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
void ServerSetTeamLoadout(uint8_t TeamIndex, struct FServerSetLoadoutTeam Params);
void ServerSetLoadoutComplete();
void ServerSetTeamLoadoutProduct(uint8_t TeamIndex, struct FLoadoutProductData ProductData);
void ServerSetTeamLoadoutPaint(struct FLoadoutTeamPaint Paint);
bool VerifyTeamIndex(uint8_t TeamIndex);
void ReplicateLoadoutToServer(struct FServerSetLoadoutParams Params);
void OnSplitScreenStatusChanged();
void HandleLocalPlayerLeave(class ULocalPlayer* OldPlayer);
void HandleLocalPlayerJoin(class ULocalPlayer* NewPlayer);
void ServerSplitScreenStatusChanged(unsigned long bInSplitScreen);
void UpdateSplitScreenStatus();
void OnPartyLeaderChanged();
void ServerSetPartyLeader(struct FUniqueNetId InPartyLeader);
void SetPartyLeader(struct FUniqueNetId InPartyLeader);
void HandlePartyChanged(class UOnlineGameParty_X* Party);
void UpdatePartyStatus();
void OnSkillTierChanged();
void ReplicateSkillTier(struct FUpdatedPlayerSkillRating NewSkillRating);
void OnTitleChanged();
void SetTitle(struct FName NewTitle);
```

```
void HandlePlayerTitle(class UOnlineGamePlayerTitles_TA* TitlesObj, struct FUniqueNetId  
InPlayerId);  
void SyncPlayerTitle();  
void UpdateTitleFromLoadout();  
void UpdateTitle();  
void HandleCameraChanged(class ACameraSettingsActor_TA* Actor);  
void SetCamera(class ACameraSettingsActor_TA* Actor);  
void eventOnOwnerChanged();  
void GetNewFriendKey();  
void SetPlayerHistoryKey(TArray<uint8_t> HistoryKeyArray);  
void ServerSetDistracted(unsigned long Value);  
void ServerSetPlayerHistoryKey(uint8_t HistoryKeyArray);  
void eventOnNewPlayerHistoryKey();  
void AddPlayerHistoryKey(class UOnlineRecentPlayersList* PlayersList);  
class FString GetSanitizedPlayerName();  
void HandlePlayerNameChanged(class APRI_X* InPRI);  
void OnRep_UniqueId();  
void ClientInitialize(class AController* C);  
void OnBotBannerProductIDChanged();  
void OnBotAvatarProductIDChanged();  
void eventPostBeginPlay();  
void eventReplicatedEvent(struct FName VarName);  
void EventPlayerGoalExplosionChanged(class APRI_TA* PRI);  
void EventCurrentVoiceRoomChanged(class APRI_TA* PRI);  
void EventIdleBannedChanged(class APRI_TA* PRI);  
void EventQuitSeverityChanged(class APRI_TA* PRI);  
void EventOwnerChanged(class APRI_TA* PRI);  
void EventSpectatorShortcutChanged(class APRI_TA* PRI);  
void EventServerUnlockedAchievement(class APRI_TA* PRI, int32_t AchievementId, uint8_t  
AchievementType);  
void EventMatchAdmin(class APRI_TA* PRI);  
void EventReplacingBotChanged(class APRI_TA* PRI);  
void EventScoredGoal(class APRI_TA* PRI, struct FVector BallHitLocation);  
void EventServerChangeTeamFailed(int32_t TeamIndex);  
void EventSkillTierChanged(class APRI_TA* PRI);  
void EventTitleChanged(class APRI_TA* PRI);  
void EventStartVoteToForfeitDisabledChanged(class APRI_TA* PRI);  
void EventCommittedMatchStats(class APRI_TA* PRI);  
void EventCommittedProductStats(class APRI_TA* PRI);  
void EventPawnTypeChanged(class APRI_TA* PRI);  
void EventStatTitlesSet(class APRI_TA* PRI);  
void EventWonMVP(class APRI_TA* PRI);  
void EventScorePoint(class APRI_TA* PRI, int32_t Points);  
void EventPartyLeaderChanged(class APRI_TA* PRI);  
void EventCameraChanged(class APRI_TA* PRI);  
void EventPersistentCameraSet(class APRI_TA* PRI);  
void EventDistracted(class APRI_TA* PRI);  
void EventSplitScreenStatusChanged(class APRI_TA* PRI);  
void EventCarSet(class APRI_TA* PRI);  
void EventCarPreUpdate(class APRI_TA* PRI);  
void EventReadyChanged(class APRI_TA* PRI);  
void EventInvalidPsyponixID();  
void EventStatTickerMessage(class APRI_TA* Receiver, class APRI_TA* Victim, class  
UStatEvent_TA* StatEvent);
```

```

void EventVanityChanged(uint8_t InVanityType);
void EventSelectedLoadout(class APRI_TA* PRI);
void EventRequestedLoadout(struct FLoadoutData& OutLoadout, struct
FLoadoutAttributesArray& OutAttributes);
void EventStatEvent(class APRI_TA* PRI, class UStatEvent_TA* StatEvent, int32_t Count);
void EventGameEventChanged(class APRI_TA* PRI);
};

// Class TAGame.__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1 : public UObject
{
public:
class AGameEvent_Soccar_TA*           GameEvent;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ConsecutiveMatchTracker_TA__HandleGameEnded_0x1");
}

return uClassPointer;
};

bool __ConsecutiveMatchTracker_TA__HandleGameEnded_0x1(struct FMatchData Player);
};

// Class
TAGame.___GFxData_Clubs_TA_SetClubInvites_0x1___GFxData_Clubs_TA_SetClubInvites_0x
1_0x1
// 0x0008 (0x0060 - 0x0068)
class
U___GFxData_Clubs_TA_SetClubInvites_0x1___GFxData_Clubs_TA_SetClubInvites_0x1_0x1 :
public UObject
{
public:
class UClubInvite_X*           ClubInvite;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.___GFxData_Clubs_TA_SetClubInvites_0x1___GFxData_Clubs_TA_SetClubInvites_0x1

```

```

    _0x1");
}

return uClassPointer;
};

bool
__GFxData_Clubs_TA_SetClubInvites_0x1__GFxData_Clubs_TA_SetClubInvites_0x1_0x1(class UClubInviteNotification_TA* N);
};

// Class TAGame.ClubInviteNotification_TA
// 0x0070 (0x0170 - 0x01E0)
class UClubInviteNotification_TA : public UNotification_TA
{
public:
    struct FUniqueNetId           PlayerID;           // 0x0170 (0x0048)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    uint64_t                      ClubID;             // 0x01B8 (0x0008)
    [0x0000000040000000] (CPF_EditInlineNotify)
    class FString                 ClubName;           // 0x01C0 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString                 ClubTag;            // 0x01D0 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ClubInviteNotification_TA");
        }
    }

    return uClassPointer;
};

bool ShouldShow();
struct FUniqueNetId GetSenderID();
class UClubInviteNotification_TA* SetClubDetails(uint64_t InClubID, class FString InClubName,
class FString InClubTag);
class UClubInviteNotification_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.__GFxData_Clubs_TA_SetClubInvites_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxData_Clubs_TA_SetClubInvites_0x1 : public UObject
{
public:
    TArray<class UClubInvite_X*>           Invites;           // 0x0060 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Clubs_TA_SetClubInvites_0x1");
}

return uClassPointer;
};

void
__GFxData_Clubs_TA_SetClubInvites_0x1__GFxData_Clubs_TA_SetClubInvites_0x1_0x3(class UClubInviteNotification_TA* Notification);
void
__GFxData_Clubs_TA_SetClubInvites_0x1__GFxData_Clubs_TA_SetClubInvites_0x1_0x2(class UClubInviteNotification_TA* Notification);
void __GFxData_Clubs_TA_SetClubInvites_0x1();
};

// Class
TAGame.__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProduct
Quantity_0x3_0x1 ProductAsset_Logo_TA
// 0x00038 (0x01060 - 0x006138)
class
U__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProductQuantity
_0x3_0x1 : public UObject
{
public:
class UOnlineProduct_TA*ProductAsset_Logo_TA : public UProductAsset_TA
{
public:
class UTexture* Texture; // 0x0100 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor TeamColor; // 0x0108 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor Op; CustomColor; // 0x060118
(0x00108) [0x0000000000000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProductQ
uantity_0x3_0x1");
}

return uClassPointer;
};

```

```
bool  
____SaveData_TA__GetOnlineProductQuantity_0x2____SaveData_TA__GetOnlineProductQuantity_0  
x3_0x1(struct FProfileProduct PP);  
};  
  
// Class TAGame.Profile(CPF_Edit)  
TArray<class UProduct_TA*> Bots2; // 0x0128 (0x0010)  
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Logo_TA");  
}  
  
return uClassPointer;  
};  
  
struct FLinearColor GetFontColor(unsigned long bSwapColors);  
TArray<struct FLinearColor> GetStadiumColors(unsigned long bSwapColors);  
class FString GetTeamName();  
};  
  
// Class TAGame.ProductAttribute_InheritCarSetting_TA  
// 0x009810 (0x00680 - 0x00F890)  
class UProfile_TA : public UObject  
{  
public:  
int32_t LocalId; ductAttribute_InheritCarSetting_TA : public  
UProductAttribute_TA  
{  
public:  
TArray<struct FInheritedParameter> Parameters; // 0x00680  
(0x00104) [0x00000000000000400001]  
class FString ProfileName; (CPF_Edit |  
CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_InheritCarSetting_TA");  
}  
  
return uClassPointer;  
};
```

```

void ApplySettings(struct FName DestinationParamName, struct FName ParentParamName,
class UMeshComponent* Parent, class UISetParameter* Child);
void Inherit(class UMeshComponent* Parent, class UObject* ChildObject);
};

// Class TAGame.ProductStat_TA
// 0x006850 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class USaveData_TA*60 - 0x00B0)
class UProductStat_TA : public UObject
{
public:
class FString SaveDataLabel; // 0x007860 (0x00108)
[0x0000000000408002000] (CPF_Transient)
int32_tConst | CPF_Localized | CPF_NeedCtorLink)
class FString ControllerID;Description; // 0x00870
(0x00104) [0x0000000000408002000] (CPF_Transient)
unsigned long bShouldRandomizeLoadoutEachGame : 1; // 0x0084
(0x0004) [0x0000000000000000] [0x00000001]
int32_tConst | CPF_Localized | CPF_NeedCtorLink)
class APRI_TA* PRI; ProfileVersion; // 0x00880
(0x00048) [0x000000008000002000] (CPF_Transient)
int32_t EditLoadoutSetIndex;ProductID; // 0x008C8
(0x0004) [0x00000008000002000] (CPF_Transient)
class UObjectProvider*ACar_TA* SaveObjectProvider; Car;
/ 0x0090 (0x0008) [0x00000080000040820080] (CPF_ExportObject | CPF_Transient |
CPF_Component | CPF>EditInline)
TArray<class USaveObject_TA*>Transient)
class AGameEvent_TA* SaveObjects;GameEvent; // 0x0098
(0x00108) [0x0000008000004002000] (CPF_NeedCtorLink)
class UProfileLoadoutSaveTransient)
class AGameEvent_Soccer_TA* LoadoutSav SoccerGame; //
0x00A80 (0x0008) [0x00000008000002000] (CPF_Transient)
struct FScriptDelegateint32_t
_EventPreEquipProductToLoadouts__Delegate; // 0x00B0 (0x0018) [0x0000000000400000]
(CPF_NeedCtorLink)
struct FScriptDelegate _EventActiveBindingsChanged__Delegate;
Value; // 0x00CA8 (0x001804) [0x0000004000004002000]
(CPF_NeedCtorLink)
struct FScriptDelegate _EventProfileProductChanged__Delegate; Transient)
unsigned long bOnlyMinorUnitsInMetricDistanceStrings : 1; // 0x00E0AC
(0x001804) [0x0000000000404002] [0x00000001] (CPF_NeedCtorLinkConst | CPF_Config)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfileProductStat_TA");
}

return uClassPointer;
}

```

};

void \_\_Profile\_TA\_\_CommonInit\_0x5(class UPlayerVanitySave\_TA\* LoadedVanitySave);  
void \_\_Profile\_TA\_\_CommonInit\_0x7();  
void \_\_Profile\_TA\_\_CommonInit\_0x6(struct FProductInstanceId \_);  
void \_\_Profile\_TA\_\_CommonInit\_0x2(class UProfileLoadoutSave\_TA\* Save);  
void \_\_Profile\_TA\_\_CommonInit\_0x4();  
void \_\_Profile\_TA\_\_CommonInit\_0x3();  
void \_\_Profile\_TA\_\_CommonInit\_0x1(class USaveObject\_TA\* S);  
void \_\_Profile\_TA\_\_Unload\_0x1(class USaveObject\_TA\* S);  
bool \_\_Profile\_TA\_\_GetUniqueNetId\_0x1(class ULocalPlayer\* P);  
class USaveObject\_TA\* \_\_Profile\_TA\_\_In static class FString GetDistanceString(class APlayerControllerBase\_TA\* ForPC, int32\_t Meters);  
stantiateSaveObjects\_0x2(class UClass\* T);  
bool \_\_Profile\_TA\_\_InstantiateSaveObjects\_0x1(class UClass\* T);  
void \_\_Profile\_TA\_\_OnProfileSet\_0x1(class USaveObject\_TA\* T);  
bool \_\_Profile\_TA\_\_GetLocalPlayer\_0x1(class ULocalPlayer\* P);  
bool IsProductEquipped(struct FProductHashID HashID);  
void UpdateSaveObjectReference(class USaveObject\_TA\* NewObject);  
class ULocalPlayer\* GetLocalPlayer();  
void OnProfileSet();  
TArray<class UClass\*> InstantiateSaveObjects();  
struct FUniqueNetId GetUniqueNetId();  
struct FProductInstanceId GetEquippedTitleInstanceId();  
TArray<struct FProfileProduct> GetProfileProducts();  
void PrintDebugInfo(class UDebugDrawer\* Drawer);  
void ResetControlBindingsToPreset c FString GetTimeString(int32\_t Seconds);  
static class FString GetDisplayValues(class UControlPreset\_X\* ControlPreset);  
void ResetAllSavedControlBindings();  
void SetControlsPreset(struct FName PresetName);  
static void AddLoadoutProductData(int32\_t SlotIndex, int32\_t ProductID, class UOnlineProduct\_TA\* OnlineProduct, TArray<class UProductAttribute\_TA\*> Attributes, TArray<struct FLoadoutProductData>& OutProducts);  
void InitServerSetLoadoutParams(struct FLoadoutData Loadouts, TArray<struct FProfileProduct> ProfileProducts, struct FServerSetLoadoutParams& OutParams);  
struct FServerSetLoadoutParams BuildServerSetLoadoutParams();  
void SetEditingLoadoutSet(int32\_t Index);  
class ULoadoutSet\_TA\* GetEditingLoadoutSet();  
class ULoadout\_TA\* GetEditingLoadout(int32\_t TeamIndex);  
void SetCustomMatchSettings(uint8\_t Type, struct FCustomMatchSettings& Settings);  
struct FCustomMatchSettings GetCustomMatchSettings(uAPlayerControllerBase\_TA\* ForPC, int32\_t StatValue);  
static class FString GetDescription();  
static class FString GetLabel();  
void AddStatValue(int832\_t Type);  
class UGameTags\_TA\* GetGameTags(uint8\_t Type);  
void ValidateLoadoutSet(class ULoadoutSet\_TA\* InLoadoutSet);  
void ValidateLoadoutsWithProduct(struct FProductInstanceId OnlineID);  
void ValidateLoadouts();  
void OnLoaded(class USaveData\_TA\* InSaveData);  
void Save();  
void Unload();  
void ValidateVanityObjects();  
void HandleRemovedOnlineProduct(class USaveData\_TA\* PlayerSaveData, class

```

UOnlineProduct_TA* OnlineProduct);
void CommonInit();
void Init(class USaveData_TA* InSaveData, class FString InProfileName, int32_t InLocalID);
void EventProfileProductChanged();
void EventActiveBindingsChanged(class UProfile_TA* Profile);
void EventPreEquipProductToLoadouts(class UProfile_TA* Profile, int32_t ProductID);
};

// Class TAGame._SaveData_TA__GetOnlineProductQuantity_0x2
// 0x0020 (0x0060 - 0x0080)
class U_SaveData_TA__GetOnlineProductQuantity_0x2 : public UObject
{
public:
TArray<struct FProductInstanceID> EquippedOnlineProducts; // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProfileProduct> EquippedProfileProducts; // 0x0070
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._SaveData_TA__GetOnlineProductQuantity_0x2");
}

return uClassPointer;
};

bool __SaveData_TA__GetOnlineProductQuantity_0x3(class UOnlineProduct_TA* Op);
bool __SaveData_TA__GetOnlineProductQuantity_0x2(class UOnlineProduct_TA* OpAmount);
void SetCar(class ACar_TA* InCar);
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void HandleCarDestroyed(class APawn_X* P);
void HandleCarRefSet(class APRI_TA* InPRI);
void HandleGameEventChanged(class APRI_TA* InPRI);
void ShutDown();
void OnInit();
void Init(class APRI_TA* InPRI, int32_t InProductID);
};

// Class TAGame.Car_TA
// 0x02A0B8 (0x08A8 - 0x0B4860)
class ACar_TA : public AVehicle_TA
{
public:
TArray<class ACarComponent_TA*> DefaultCarComponents; // 0x08A8
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UEngineAudioComponent_TA* EngineAudio; // 0x08B8
(0x0008) [0x00000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_EditInline)

```

```

class UEngineAudioREVComponent_TA*           EngineAudioRev;          // 0x08C0
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF_EditInline)
class UThrottleShakeComponent_TA*           ThrottleShake;         // 0x08C8
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | 
CPF_EditInline)
class ACarComponent_FlipCar_TA*             FlipComponent;        // 0x08D0
(0x0008) [0x0000000000002000] (CPF_Transient)
uint8_t                      DemolishTarget;       // 0x08D8 (0x0001)
[0x0000000000002000] (CPF_Transient)
uint8_t                      DemolishSpeed;        // 0x08D9 (0x0001)
[0x0000000000002000] (CPF_Transient)
class UProductLoader_TA*                   Loadout;              // 0x08E0 (0x0008)
[0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
unsigned long                  bLoadoutSet : 1;      // 0x08E8 (0x0004)
[0x0000004000000000] [0x00000001]
unsigned long                  bDemolishOnOpposingGround : 1; // 0x08E8
(0x0004) [0x0000004000002000] [0x00000002] (CPF_Transient)
unsigned long                  bWasOnOpposingGround : 1; // 0x08E8 (0x0004)
[0x0000004000002000] [0x00000004] (CPF_Transient)
unsigned long                  bDemolishOnGoalZone : 1; // 0x08E8 (0x0004)
[0x0000004000002000] [0x00000008] (CPF_Transient)
unsigned long                  bWasInGoalZone : 1;      // 0x08E8 (0x0004)
[0x0000004000002000] [0x00000010] (CPF_Transient)
unsigned long                  bOverrideHandbrakeOn : 1; // 0x08E8 (0x0004)
[0x0000000000002020] [0x00000020] (CPF_Net | CPF_Transient)
unsigned long                  bTeamBumpsUseCarForceMultiplier : 1; // 0x08E8
(0x0004) [0x0000000000000000] [0x00000040]
unsigned long                  bOverrideBoostOn : 1;      // 0x08E8 (0x0004)
[0x0000000000002020] [0x00000080] (CPF_Net | CPF_Transient)
unsigned long                  bUseDefaultLoadout : 1; // 0x08E8 (0x0004)
[0x0000000000004000] [0x00000100] (CPF_Config)
class AFXActor_X*                ExitFXArchetype;        // 0x08F0 (0x0008)
[0x0000000000000001] (CPF>Edit)
class AExplosion_X*              DemolishExplosionArchetype; // 0x08F8
(0x0008) [0x0000000000000001] (CPF>Edit)
class AExplosion_X*              DemolishExplosion;        // 0x0900 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UClass*                   TargetClass;            // 0x0908 (0x0008)
[0x0000000000000002] (CPF_Const)
class UTargetIndicator_TA*       TargetIndicatorArchetype; // 0x0910
(0x0008) [0x0000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | 
CPF>EditInline)
TArray<class UTargetIndicator_TA*>   TargetIndicators;        // 0x0918
(0x0010) [0x0000000004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | 
CPF_NeedCtorLink | CPF>EditInline)
float                         MaxTimeForDodge;        // 0x0928 (0x0004)
[0x0000000000000001] (CPF>Edit)
float                         LastWheelsHitBallTime; // 0x092C (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FLoadoutTeamPaint        TeamPaint;              // 0x0930 (0x0010)
[0x0000008100002020] (CPF_Net | CPF_Transient)
struct FClubColorSet             ClubColors;            // 0x0940 (0x0008)
[0x0000008100002020] (CPF_Net | CPF_Transient)

```

```
float ReplicatedCarScale; // 0x0948 (0x0004)
[0x0000000100000020] (CPF_Net)
struct FDemolishData ReplicatedDemolish; // 0x0950 (0x0028)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FDemolishData2 ReplicatedDemolish_CustomFX; // 0x0978
(0x0030) [0x0000004000002000] (CPF_Transient)
struct FDemolishDataGoalExplosion ReplicatedDemolishGoalExplosion; //
0x09A8 (0x0030) [0x0000004100002020] (CPF_Net | CPF_Transient)
class AFXActor_X* BodyFXActor; // 0x09D8 (0x0008)
[0x0000004000002000] (CPF_Transient)
class APRI_TA* AttackerPRI; // 0x09E0 (0x0008)
[0x0000000000000000]
struct FVector MouseAccel; // 0x09E8 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector MouseAirAccel; // 0x09F4 (0x000C)
[0x0000000000002000] (CPF_Transient)
class ASpecialPickup_TA* AttachedPickup; // 0x0A00 (0x0008)
[0x0000000100002020] (CPF_Net | CPF_Transient)
class ARumblePickups_TA* RumblePickups; // 0x0A08 (0x0008)
[0x0008000000000020] (CPF_Net)
struct FVector ReplayFocusOffset; // 0x0A10 (0x000C)
[0x0000000000002000] (CPF_Transient)
float AddedBallForceMultiplier; // 0x0A1C (0x0004)
[0x000000000000021] (CPF_Edit | CPF_Net)
float AddedCarForceMultiplier; // 0x0A20 (0x0004)
[0x000000000000021] (CPF_Edit | CPF_Net)
class UCarTrajectoryComponent_TA* CarTrajectoryComponent; // 0x0A28
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class AGameEvent_TA* GameEvent; // 0x0A30 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UNameplateComponentCar_TA* NameplateComponentCar; // 0xA38 (0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UExplosionHitHandler_TA* ExplosionHitHandler; // 0x0A40
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float ReplicatedCarMaxLinearSpeedScale; // 0x0A48 (0x0004)
[0x0000000100000020] (CPF_Net)
struct FName PostMatchAnim; // 0x0A4C (0x0008)
[0x0000000000000020] (CPF_Net)
struct FScriptDelegate __EventVehicleSetup__Delegate; // 0x0A58
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventBumpedCar__Delegate; // 0x0A70
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventDemolished__Delegate; // 0x0A88
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTeamSet__Delegate; // 0x0AA0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTeamPaintChanged__Delegate; // 0x0AB8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventHitBall__Delegate; // 0x0AD0 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventLanded__Delegate; // 0x0AE8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSuperSonicChanged__Delegate; // 0xB00
```

```
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventHitWorld__Delegate;           // 0x0B18
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPreparingDemoFX__Delegate;   // 0x0B30
(0x0018) [0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __RumblePickups__ChangeNotify;    // 0x0B3048
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Car_TA");
}

return uClassPointer;
};

void __Car_TA__HandleAssetLoaded_0x1(class UObject* Mutator);
void __Car_TA__RefreshTargetIndicators_0x1(class UTargetIndicator_TA* TargetIndicator);
void __RumblePickups__ChangeNotifyFunc();
void SetCarMaxLinearSpeedScale(float InMaxLinearSpeedScale);
bool IsAffectedByExplosionHitHandler();
void SetExplosionHitHandler(class UExplosionHitHandler_TA* Handler);
void SetAttachedPickupForLegacyReplay(class ASpecialPickup_TA* NewPickup);
float GetMaxDriveBackwardsSpeed();
float GetMaxDriveForwardSpeed();
struct FVector GetReplayFocusLocation();
void EnablePodiumMode();
void ServerEquipProduct(struct FName ProductName);
void PrintDebugInfo(class UDebugDrawer* Drawer);
static class UObject* GetOutermostObject(class UObject* InObject);
void CopyPushFactorCurve();
void ClearAttackerPRI();
void SetAttackerPRI(class APRI_TA* Attacker);
void HandleTargetRemoved(class UTarget_TA* Target);
void HandleTargetAdded(class UTarget_TA* Target);
void RefreshTargetIndicators();
class FString GetReplicatedLoadoutString();
void eventOnSuperSonicChanged();
void eventOnGroundChanged();
void FellOutOfWorld();
class UNetConnection* GetNetConnection();
class APlayerController* GetTopPlayerController();
void DemolishDestroyTimer();
void ClearFX();
void SetDemoFXArchetypeOverrides(class AFXActor_X* OutDemoFXArchetype);
bool IsDemolished();
void FinishDemo(struct FDemolishData Data, class AFXActor_X* ExplosionFxArchetype);
void QueueDemolish(struct FDemolishData Data, class APRI_TA* GoalExplosionOwner);
void QueueDemolish_Deprecated(struct FDemolishData Data, class AFXActor_X*
```

```
CustomDemoFX);
void OnDemolishedGoalExplosion(struct FDemolishDataGoalExplosion Data);
void OnDemolished_CustomFX(struct FDemolishData2 Data);
void Demolish(class ARBActor_TA* Demolisher);
void GoalExplosionDemolish(class APRI_TA* ScoringPlayer);
void SetSoundMode(struct FName NewModeValue, struct FName NewModeGroup);
bool Teleport(struct FVector SpawnLocation, struct FRotator SpawnRotation, unsigned long bStopVelocity, unsigned long bUpdateRotation, float ExtraForce);
void OnTeleport(class USeqAct_Teleport* Action);
bool HasDodgeInput();
void OnJumpReleased();
void OnJumpPressed();
void eventSetVehicleInput(struct FVehicleInputs NewInput);
bool CanDemolish(class ACar_TA* HitCar);
bool ShouldDemolish(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal, uint8_t& Result);
void InitTimeOfImpactFromOldRBState(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal, struct FTimeOfImpactData& Impact);
void AddBumpImpulse(struct FVector BumpImpulse, class ACar_TA* BumpingCar);
void BumpCar(float Speed, class ACar_TA* OtherCar, struct FVector HitLocation, struct FVector HitNormal);
uint8_t ApplyCarImpactForces(class ACar_TA* OtherCar, struct FVector HitLocation, struct FVector HitNormal);
bool IsValidImpactNormalHit(float DotProductTolerance, struct FTimeOfImpactData& Impact);
struct FVector GetBumpImpulse(float Speed, class ACar_TA* OtherCar, struct FVector HitLocation, struct FVector HitNormal);
bool IsBumperHit(class ACar_TA* OtherCar, struct FVector HitLocation, struct FVector HitNormal);
void ApplyBallImpactForces(class ABall_TA* Ball, struct FVector HitLocation);
bool IsDodging();
void OnHitBall(class ABall_TA* Ball, struct FVector HitLocation, struct FVector HitNormal);
void ApplyHitWallAdjustmentForces(struct FAccumulatedRigidBodyCollision Collision, struct FVector HitLocation, struct FVector HitNormal);
void OnTouchCarDemoActor();
void eventOnRigidBodyCollision(struct FAccumulatedRigidBodyCollision Collision);
bool AnyWheelTouchingGround();
class ACarComponent_TA* GiveCarComponent(class ACarComponent_TA* ComponentArchetype, class APRI_TA* Activator);
void AddDefaultCarComponents();
void NotifyWhenVehicleSetup(struct FScriptDelegate Callback);
void HandleVisualAssetLoaded(class UProductAsset_TA* Asset);
void DetachPrimitiveComponent(class UPrimitiveComponent* Component);
void HandleWheelContactChanged(class UWheel_TA* Wheel);
void OnVehicleSetup();
void HandleAllAssetsLoaded(class UProductLoader_TA* Loader);
void HandlePostMatchCelebration(class UAssetAttribute_PostMatchCelebration_TA* PostMatchCelebration);
void HandleAssetLoaded(class UProductLoader_TA* Loader, class UProductAsset_TA* Asset);
void RespawnInPlace();
void SetCarScale(float NewScale);
void OnClubColorsChanged();
void HandleTeamChanged(class APRI_X* MyPRI);
bool UpdateTeamLoadout();
void InitTeamPaint();
```

```

int32_t GetLoadoutTeamIndex();
int32_t GetPreviewTeamIndex();
bool HasTeam();
void HandleLoadoutSelected(class APRI_TA* MyPRI);
void SetLoadout(struct FClientLoadoutData& Data);
void HandleGameStateChanged(class AGameEvent_TA* InGameEvent);
void HandleGameEventChanged(class APRI_TA* MyPRI);
void OnPRIChanged();
void OnControllerChanged();
void eventDestroyed();
void CreateRumblePickups();
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventPreparingDemoFX(struct FDemolishData DemoData, class AExplosion_X*
DemoExplosion);
void EventHitWorld(class ACar_TA* Car, class AActor* HitActor, struct FVector HitNormal);
void EventSuperSonicChanged(class ACar_TA* Car);
void EventLanded(class ACar_TA* Car);
void EventHitBall(class ACar_TA* Car, class ABall_TA* Ball, struct FVector HitLocation, struct
FVector HitNormal);
void EventTeamPaintChanged(class ACar_TA* Car);
void EventTeamSet(class ACar_TA* Car);
void EventDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void EventBumpedCar(class ACar_TA* Car, class ACar_TA* HitCar, struct FVector HitLocation);
void EventVehicleSetup(class ACar_TA* Car);
};

// Class
TAGame.ProductAsset_Logo_TA____GFxData_Mutators_TA_SetGameModeMutatorPresets_0x2
____GFxData_Mutators_TA_SetGameModeMutatorPresets_0x2_0x1
// 0x003814 (0x00F860 - 0x0130074)
class UProductAsset_Logo_TA : public
UProductAsset_TA____GFxData_Mutators_TA_SetGameModeMutatorPresets_0x2____GFxData_M
utators_TA_SetGameModeMutatorPresets_0x2_0x1 : public UObject
{
public:
struct FCategorySettingPair P; // 0x0060 (0x0014)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame____GFxData_Mutators_TA_SetGameModeMutatorPresets_0x2____GFxData_Mutators_
TA_SetGameModeMutatorPresets_0x2_0x1");
}
{
public:
class UTexture* return uClassPointer;
};

```

```

bool
__GFxData_Mutators_TA_SetGameModeMutatorPresets_0x2__GFxData_Mutators_TA_SetG
ameModeMutatorPresets_0x2_0x1(struct FCustomGameSetting S);
};

// Class TAGame.GFxData_Mutators_TA
// 0x0098 (0x0098 - 0x0130)
class UGFxData_Mutators_TA : public UGFxDataSingleton_X
{
public:
struct FName           Texture;      SelectedMapName;          // 0x00F98
(0x0008) [0x00000000040000001] (CPF_Edit)
struct FLinearColor     TeamColor | CPF_EditInlineNotify
int32_t                 SelectedGameMode;        // 0x00A0 (0x0004)
[0x0000000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t                 SelectedBotDifficulty; // 0x00A4 (0x0004)
[0x0000000040000001] (CPF_Edit | CPF_EditInlineNotify)
TArray<struct FGameModeData>   GameModes;       // 0x01000A8
(0x0010) [0x00000000040400001] (CPF_Edit)
struct FLinearColor     CustomColor;    | CPF_NeedCtorLink |
CPF>EditInlineNotify)
TArray<struct FCustomGameSetting> CustomGameSettings; // 0x00B8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<class UGameSettingCategory_X*> GameSettingCategories; // 0x01100C8 (0x0010) [0x000000000004020001] (CPF>Edit)
TArray<class UProduct_TA*> Transient | CPF_NeedCtorLink)
TArray<class UGameSetting_X*> GameModeMutatorSettingPresets; // 0x00D8 (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                 PresetCategoryIndex; // 0x00E8 (0x0004)
[0x0000004000000000]
TArray<int32_t>         Bots2     MutatorGroupIDs; // 0x00F0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FModeMapPair> ModeMaps; // 0x01200 (0x0010)
(0x0010) [0x00000000004020001] (CPF>EditTransient | CPF_NeedCtorLink)

uint8_t                 SettingsType; // 0x0110 (0x0001)
[0x0000000000002000] (CPF_Transient)
class UProfile_TA*       Profile;    // 0x0118 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGameMode_TA*      GameModeObj; // 0x0120 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGameTags_TA*      GameTags;   // 0x0128 (0x0008)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_LogoGFxData_Mutators_TA");
}
}

```

```

}

return uClassPointer;
};

struct FLinearColor GetFontColor(unsigned long bSwapColors);
TArray<struct FLinearColor> GetStadiumColors(unsigned long bSwapColors);
class FString GetTeamName();
};

// Class TAGame.ProductAttribute_InheritCarSetting
bool __GFxData_Mutators_TA__SetGameModeMutatorPresets_0x3(struct FCustomGameSetting S);
bool __GFxData_Mutators_TA__SetGameModeMutatorPresets_0x1(class UGameSetting_X* GMP);
void __GFxData_Mutators_TA__ClearPreviousGameModePresets_0x2(struct FCategorySettingPair P);
bool __GFxData_Mutators_TA__ClearPreviousGameModePresets_0x1(class UGameSetting_X* GMP);
int32_t GetMutatorGroupCount();
class FString GetLocalizedMutatorGroupLabel(int32_t GroupId);
TArray<struct FCustomGameSetting> GetMutatorsByGroupID(int32_t GroupId);
void SetCustomGameSetting(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex);
bool ResetInvalidSettings(int32_t MutatorCategoryIndex);
bool IsValidSetting(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex, unsigned long bCoerceMap);
class FString GetCustomGameSettingDescription(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex);
void ForceClearAllTags();
void SetSelectedBotDifficulty(int32_t BotDifficulty);
class FString GetGameTagValueFromIndex(class FString GameTagCategory, int32_t Index);
int32_t GetGameTagIndex(class FString GameTagCategory, class FString Value);
class FString GetSelectedGameModeDescription();
class FString GetSelectedGameModeName();
void SetSelectedGameMode(int32_t GameMode);
void SetSelectedMapName(struct FName MapName);
void InitCustomGameSettings();
void RefreshCustomGameSettings();
void ClearPreviousGameModePresets();
bool SetGameModeMutatorPresets();
void OnUpdatedSettings();
void PopulateGameModes();
void SetCustomMatchSettings(struct FCustomMatchSettings& InSettings);
struct FCustomMatchSettings GetCustomMatchSettings();
void SetGameTag(class FString Key, class FString Value);
class FString GetGameTag(class FString Key);
void InitMutatorsFromTournament(uint64_t TournamentID);
void InitMutators(uint8_t InSettingsType);
void eventOnShellSet();
class UOnlineGame_TA* GetOnlineGame();
};

// Class TAGame.GameTags_TA
// 0x001028 (0x00860 - 0x009088)
class UProductAttribute_InheritCarSetting_TA : public UProductAttribute_TA
TAGameTags_TA :

```

```

public UObject
{
public:
TArray<struct FInheritedParameterKeyValuePair>           Parameters;
Tags;                                     // 0x00860 (0x0010) [0x000800400040040001] (CPF_Edit | NeedCtorLink)
struct FScriptDelegate           _Tags_ChangeNotify;          // 0x0070 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_InheritCarSettingGameTags_TA");
}

return uClassPointer;
};

void ApplySettings(struct FName DestinationParamName, struct FName ParentParamName,
class UMeshComponent* Parent, class UISetParameter* Child);
void Inherit(class UMeshComponent* Parent, class UObject*
ChildObject__Tags__ChangeNotifyFunc());
static void ConvertToGameTags(class UObject* InOuter, TArray<class UGameTags_TA*>&
GameTags, TArray<struct FKeyValuePair>& Pairs);
void RemoveValue(class FString Value);
class FString CreateGameTagsStr();
void SetValue(class FString Key, class FString Value);
class FString GetValue(class FString Key);
void ResetTags();
void InitTags(class FString AllTags);
};

// Class TAGame.ProductStatfile_TA
// 0x005098 (0x0060 - 0x00B0F8)
class UProductStatfile_TA : public UObject
{
public:
class FStringint32_t           Label; ocallId;          // 0x0060
(0x00104) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)000000

class FString                  DescriptionProfileName; // 0x007068
(0x0010) [0x00000000004080002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class APRIUSaveData_TA*         PRI;   SaveData;          // 0x00780
(0x0008) [0x00000080000002000] (CPF_Transient)
int32_t                         ProductID; ControllerId; // 0x00880 (0x0004)
[0x00000080000002000] (CPF_Transient)
class ACar_TA*unsigned long     Car;
bShouldRandomizeLoadoutEachGame : 1;      // 0x0084 (0x0004) [0x0000000000000000]
[0x00000001]

```

```

int32_t ProfileVersion; // 0x009088 (0x00084)
[0x000000800000002000] (CPF_Transient)
class AGameEvent_TA*
int32_t GameEvent; EditLoadoutSetIndex; // 0x0098C
(0x00084) [0x00000080000002000] (CPF_Transient)
class AGameEvent_Soccar_TA* UObjectProvider* SoccarGame;
aveObjectProvider; // 0x00A90 (0x0008) [0x00000080000040820008]
(CPF_Transient)
int32_t ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
TArray<class USaveObject_TA*> SaveObjects; Value; // 0x0098
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UProfileLoadoutSave_TA* LoadoutSave; // 0x00A8
(0x00048) [0x00000040000002000] (CPF_Transient)
unsigned longstruct FScriptDelegate
bOnlyMinorUnitsInMetricDistanceStrings : 1; __EventPreEquipProductToLoadouts__Delegate; // 0x00B0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventActiveBindingsChanged__Delegate; // 0x00AC8 (0x000418) [0x00000000000040002] [0x400000001] (CPF_Const | CPF_ConfigNeedCtorLink)
struct FScriptDelegate __EventProfileProductChanged__Delegate; // 0x00E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStatfile_TA");
}

return uClassPointer;
};

static class FString GetDistanceString(class APlayerControllerBase_TA* ForPC, int32_t Meters);
void __Profile_TA__CommonInit_0x5(class UPlayerVanitySave_TA* LoadedVanitySave);
void __Profile_TA__CommonInit_0x7();
void __Profile_TA__CommonInit_0x6(struct FProductInstanceID _);
void __Profile_TA__CommonInit_0x2(class UProfileLoadoutSave_TA* Save);
void __Profile_TA__CommonInit_0x4();
void __Profile_TA__CommonInit_0x3();
void __Profile_TA__CommonInit_0x1(class USaveObject_TA* S);
void __Profile_TA__Unload_0x1(class USaveObject_TA* S);
bool __Profile_TA__GetUniqueNetId_0x1(class ULocalPlayer* P);
class USaveObject_TA* __Profile_TA__InstantiateSaveObjects_0x2(class UClass* T);
bool __Profile_TA__Instantic class FString GetTimeString(int32_t Seconds);
ateSaveObjects_0x1(class UClass* T);
void __Profile_TA__OnProfileSet_0x1(class USaveObject_TA* T);
bool __Profile_TA__GetLocalPlayer_0x1(class ULocalPlayer* P);
bool IsProductEquipped(struct FProductHashID HashID);
void UpdateSaveObjectReference(class USaveObject_TA* NewObject);
class ULocalPlayer* GetLocalPlayer();
void OnProfileSet();

```

```
 TArray<class UClass*> Instantic class FString GetDisplayateSaveObjects();
struct FUniqueNetId GetUniqueNetId();
struct FProductInstanceID GetEquippedTitleInstanceID();
TArray<struct FProfileProduct> GetProfileProducts();
void PrintDebugInfo(class UDebugDrawer* Drawer);
void ResetControlBindingsToPresetValues(class APlayerControllerBase_TA* ForPC, int32_t StatValue);
static class FString GetDescription();
static class FStrUControlPreset_X* ControlPreset();
void ResetAllSavedControlBindings();
void SetControlsPreset(struct FName PresetName);
static void AddLoadoutProductData(int32_t SlotIndex, int32_t ProductID, class UOnlineProduct_TA* OnlineProduct, TArray<class UProductAttribute_TA*> Attributes, TArray<struct FLoadoutProductData>& OutProducts);
void InitServerSetLoadoutParams(struct FLoadoutData Loadouts, TArray<struct FProfileProduct> ProfileProducts, struct FServerSetLoadoutParams& OutParams);
struct FServerSetLoadoutParams BuildServerSetLoadoutParams();
void SetEditingLoadoutSet(int32_t Index);
class ULoadoutSet_TA* GetEditingLoadoutSet();
class ULoadout_TA* GetEditingLoadout(int32_t TeamIndex);
void SetCustomMatchSettings(uint8_t Type, struct FCustomMatchSettings& Settings);
struct FCustomMatchSettings GetLabel();
void AddStatValue(int32_t Amount);
void SetCar(class ACar_TA* InCar);
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void HandleCarDestroyed(class APawn_X* P);
void HandleCarRefSet(class APRI_TA* InPRI);
void HandleGameEventChanged(class APRI_TA* InPRI);
void ShutDown();
void OnInit();
void Init(class APRI_TA* InPRI, int32_t InProductIDCustomMatchSettings(uint8_t Type));
class UGameTags_TA* GetGameTags(uint8_t Type);
void ValidateLoadoutSet(class ULoadoutSet_TA* InLoadoutSet);
void ValidateLoadoutsWithProduct(struct FProductInstanceID OnlineID);
void ValidateLoadouts();
void OnLoaded(class USaveData_TA* InSaveData);
void Save();
void Unload();
void ValidateVanityObjects();
void HandleRemovedOnlineProduct(class USaveData_TA* PlayerSaveData, class UOnlineProduct_TA* OnlineProduct);
void CommonInit();
void Init(class USaveData_TA* InSaveData, class FString InProfileName, int32_t InLocalID);
void EventProfileProductChanged();
void EventActiveBindingsChanged(class UProfile_TA* Profile);
void EventPreEquipProductToLoadouts(class UProfile_TA* Profile, int32_t ProductID);
};

// Class TAGame.__GFxData_Mutators_TA__SetGameModeMutatorPresets_0x2
// 0x0004 (0x0060 - 0x0064)
class U_GFxData_Mutators_TA__SetGameModeMutatorPresets_0x2 : public UObject
{
public:
unsigned long          bSettingsChanged : 1;           // 0x0060 (0x0004)
```

[0x0000000000000000] [0x00000001]

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__GFxDATA_Mutators_TA_SetGameModeMutatorPresets_0x2");  
}  
  
return uClassPointer;  
};  
  
void __GFxDATA_Mutators_TA_SetGameModeMutatorPresets_0x2(struct FCategorySettingPair  
P);  
};  
  
// Class  
TAGame.__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProduct  
Quantity_0x3_0x1  
// 0x0008 (0x0060 - 0x0068)  
class  
U__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProductQuantity  
_0x3_0x1 : public UObject  
{  
public:  
class UOnlineProduct_TA* Op; // 0x0060 (0x0008)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProductQ  
uantity_0x3_0x1");  
}  
  
return uClassPointer;  
};  
  
bool  
__SaveData_TA_GetOnlineProductQuantity_0x2__SaveData_TA_GetOnlineProductQuantity_0  
x3_0x1(struct FProfileProduct PP);  
};  
  
// Class TAGame.__SaveData_TA_GetOnlineProductQuantity_0x2  
// 0x0020 (0x0060 - 0x0080)
```

```
class U__SaveData_TA__GetOnlineProductQuantity_0x2 : public UObject
{
public:
TArray<struct FProductInstanceId> EquippedOnlineProducts; // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProfileProduct> EquippedProfileProducts; // 0x0070
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__SaveData_TA__GetOnlineProductQuantity_0x2");
}

return uClassPointer;
};

bool __SaveData_TA__GetOnlineProductQuantity_0x3(class UOnlineProduct_TA* Op);
bool __SaveData_TA__GetOnlineProductQuantity_0x2(class UOnlineProduct_TA* Op);
};

// Class
TAGame.__ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1__ShopNoti
ficationsManager_TA__HandleGetItemShopNotifications_0x3_0x1
// 0x0048 (0x0060 - 0x00A8)
class
U__ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1__ShopNotificatio
nsManager_TA__HandleGetItemShopNotifications_0x3_0x1 : public UObject
{
public:
struct FItemShopNotificationData ShopNotificationData; // 0x0060
(0x0048) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1__ShopNoti
ficationsManager_TA__HandleGetItemShopNotifications_0x3_0x1");
}

return uClassPointer;
};

bool
```

```

____ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1____ShopNotification
sManager_TA__HandleGetItemShopNotifications_0x3_0x1(class UNotification_TA* N);
};

// Class TAGame.ShopNotificationsManager_TA
// 0x0090 (0x0060 - 0x00F0)
class UShopNotificationsManager_TA : public UObject
{
public:
    class UItemShopNotificationsConfig_TA*           ItemShopNotificationsConfig;          // 0x0060 (0x0008) [0x00010800000002000] (CPF_Transient)
    class UGFxData_Community_TA*                     Community;                         // 0x0068 (0x0008) [0x00098000000002000] (CPF_Transient)
    class UASyncTask*                             ShopNotificationsTask;                // 0x0070 (0x0008) [0x00001000000002000] (CPF_Transient)
    TArray<struct FReadyItemShopNotification>     PendingShopNotifications;           // 0x0078 (0x0010) [0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)
    class UItemShopNotification_TA*                 CurrentNotification;               // 0x0088 (0x0008) [0x0001000000002000] (CPF_Transient)
    TArray<int32_t>                           ShownShopNotificationIDs;           // 0x0090 (0x0010) [0x00010004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FName>                      ValidMenus;                        // 0x00A0 (0x0010) [0x00010000000400000] (CPF_NeedCtorLink)
    TArray<class UClass*>                    DynamicThumbnailDisabledClasses;   // 0x00B0 (0x0010) [0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)
    struct FScriptDelegate                   __OnIconLoaded__Delegate;           // 0x00C0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate                   __Community__ChangeNotify;         // 0x00D8 (0x0018) [0x00010000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ShopNotificationsManager_TA");
        }

        return uClassPointer;
    };

    void __ShopNotificationsManager_TA__SetCommunity_0x1();
    int32_t __ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x5(struct FItemShopNotificationData A, struct FItemShopNotificationData B);
    bool __ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x4(class UNotification_TA* N);
    void __ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x2(class UItemShopNotification_TA* N);
    void __Community__ChangeNotifyFunc();
    class UGFxData_NotificationManager_TA* GetGFxNotificationManager();
    class UNotificationManager_TA* GetNotificationManager();
    void HandleSharedUIStateChanged();
}

```

```

void HandleSharedUIStateChangedDelayed();
void HandleTopMenuChanged(struct FName PrevMenu, struct FName TopMenu, unsigned long bClearingStack);
void HandleModalClosed();
void HandleShopNotificationHidden(class UItemShopNotification_TA* Notification);
void HandleShopNotificationShown(class UItemShopNotification_TA* Notification);
void ShowNotification(struct FReadyItemShopNotification ReadyShopNotification);
void ProcessPendingShopNotifications();
bool WillRetryProcessPendingShopNotifications();
bool ShouldProcessPendingNotifications();
void CheckPendingNotifications();
class FString GetNotificationTitle(struct FItemShopNotificationData NotificationData);
void HandleNotificationIconLoaded(struct FItemShopNotificationData ShopNotificationData,
class UTexture* Icon);
void HandleAssetLoaded(struct FScriptDelegate Callback, struct FAssetLoadResult& Result,
struct FItemShopNotificationData& ShopNotificationData);
void LoadNotificationIcon(struct FItemShopNotificationData ShopNotificationData, struct
FScriptDelegate Callback);
bool HasSeenShopNotification(class UItemShopNotification_TA* Notification, int32_t
ShopNotificationID);
void HandleGetItemShopNotifications(TArray<struct FItemShopNotificationData>
ShopNotifications);
void ShowItemShopNotifications();
bool ShouldShowShopNotifications();
void HandleNewsPanelToggled(unsigned long blsExpanded);
void SetCommunity(class UGFxData_Community_TA* InCommunity);
void eventConstruct();
void OnIconLoaded(struct FItemShopNotificationData ShopNotificationData, class UTexture*
Icon);
};

// Class TAGame.___ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1
// 0x0018 (0x0060 - 0x0078)
class U___ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1 : public UObject
{
public:
TArray<struct FItemShopNotificationData>      ShopNotifications;          // 0x0060
(0x0010) [0x0001000000040000] (CPF_NeedCtorLink)
class UNotificationManager_TA*           NotificationManager;        // 0x0070
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.___ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1");
}

return uClassPointer;
};

```

```

bool __ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x3(struct
FItemShopNotificationData ShopNotificationData);
bool __ShopNotificationsManager_TA__HandleGetItemShopNotifications_0x1(class
UItemShopNotification_TA* N);
};

// Class TAGame.__AchievementManager_TA__HandleTradeInComplete_0x1
// 0x0001 (0x0060 - 0x0061)
class U__AchievementManager_TA__HandleTradeInComplete_0x1 : public UObject
{
public:
    uint8_t          PrevQuality;           // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__AchievementManager_TA__HandleTradeInComplete_0x1");
        }

        return uClassPointer;
    }
};

bool __AchievementManager_TA__HandleTradeInComplete_0x1(class UOnlineProduct_TA* P);
};

// Class TAGame.AchievementManager_TA
// 0x0445C (0x008C - 0x04DE8)
class UAchievementManager_TA : public UTickableStateObject_X
{
public:
    class APlayerControllerBase_TA*      PC;           // 0x0090 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    TArray<int32_t>          UnlockedAchievements; // 0x0098 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FName>          DLC1Cars;       // 0x00A8 (0x0010)
    [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
    TArray<struct FName>          DLC2Cars;       // 0x00B8 (0x0010)
    [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
    TArray<struct FName>          ClassicCars;    // 0x00C8 (0x0010)
    [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
    TArray<struct FName>          RequiredDontLookBackProducts; // 0x00D8
    (0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
    TArray<struct FName>          RequiredAnInchAnd62MilesProducts; // 0x00E8
    (0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
    TArray<struct FName>          SARPBCArenas;   // 0x00F8 (0x0010)
    [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
    TArray<struct FName>          RequiredLevelsToPlay; // 0x0108 (0x0010)
};

```

```
[0x0000000000040000] (CPF_Const | CPF_NeedCtorLink)
class UMapSet_TA* RequiredRocketLabsArenas; // 0x0118
(0x0008) [0x0000000000000002] (CPF_Const)
TArray<struct FName> RocketLabsArenaNames; // 0x0120
(0x0010) [0x0000000000040000] (CPF_NeedCtorLink)
TArray<class UMapData_TA*> BraveTheElementsBaseLevels; // 0x0130
(0x0010) [0x0000000000040002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> RequiredRumbleActivation; // 0x0140
(0x0010) [0x0000000000040002] (CPF_Const | CPF_NeedCtorLink)
float MaxBoostTime; // 0x0150 (0x0004)
[0x0000000000000002] (CPF_Const)
float MaxTimeOnWall; // 0x0154 (0x0004)
[0x0000000000000002] (CPF_Const)
float MaxDriveDistanceKM; // 0x0158 (0x0004)
[0x0000000000000002] (CPF_Const)
float MaxHandbrakeRadius; // 0x015C (0x0004)
[0x0000000000000002] (CPF_Const)
float MinHandbrakeDotAngleForComplete; // 0x0160 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t DunksScored; // 0x0164 (0x0004)
[0x0000000000000000]
int32_t RocketBoostFills; // 0x0168 (0x0004)
[0x0000000000000000]
int32_t TotalDLC0BoostTime; // 0x016C (0x0004)
[0x0000000000000000]
int32_t GoalsScoredInCurrentGame; // 0x0170 (0x0004)
[0x0000000000000000]
int32_t SavesInCurrentGame; // 0x0174 (0x0004)
[0x0000000000000000]
int32_t AssistsInCurrentGame; // 0x0178 (0x0004)
[0x0000000000000000]
int32_t DLC2PsychoMasterExplodeCount; // 0x017C (0x0004)
[0x0000000000000000]
int32_t TotalBreakoutGoals; // 0x0180 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t MaxRocketBoostFills; // 0x0184 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t TotalBreakoutPlatformsToDamage; // 0x0188 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t TotalDLC0WheelDriveDistanceKM; // 0x018C (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t Winner_GameEventsWon; // 0x0190 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t SuperVictorious_GameEventsWon; // 0x0194 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t PitchVeteran_GameEventsPlayed; // 0x0198 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t PickMeUp_RandomItemDrops; // 0x019C (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t DropsInTheBucket_RandomItemDrops; // 0x01A0
(0x0004) [0x0000000000000002] (CPF_Const)
int32_t RidersBlock_GoalSaves; // 0x01A4 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t Savage_Goals; // 0x01A8 (0x0004)
```

```

[0x0000000000000002] (CPF_Const)
int32_t Spectacular_Goals; // 0x01AC (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t Ruthless_Goals; // 0x01B0 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t RocketGenocider_Goals; // 0x01B4 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t MadScientist_Maps; // 0x01B8 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t ComingOnStrong_GoalsOrAssists; // 0x01BC (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t PeoplePerson_CompletedMatches; // 0x01C0 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t MetaAchievementPointMax; // 0x01C4 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t BattleCarCollector_CarsCollected; // 0x01C8 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t Traveler_Maps; // 0x01CC (0x0004)
[0x0000000000000002] (CPF_Const)
unsigned long bBoostWasFull : 1; // 0x01D0 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long bPlayerInitiated : 1; // 0x01D0 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long bHandbrakePressed : 1; // 0x01D0 (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long bCheckMinuteToWinIt : 1; // 0x01D0 (0x0004)
[0x0000000000000000] [0x00000008]
unsigned long bDidInitialCheckForMinuteToWinIt : 1; // 0x01D0
(0x0004) [0x0000000000000000] [0x00000010]
unsigned long bCheckBuzzerBeater : 1; // 0x01D0 (0x0004)
[0x0000000000000000] [0x00000020]
unsigned long bDidInitialCheckForBuzzerBeater : 1; // 0x01D0
(0x0004) [0x0000000000000000] [0x00000040]
unsigned long bQualifiesForDLC1Throwback : 1; // 0x01D0
(0x0004) [0x0000000000000000] [0x00000080]
unsigned long bSSFuryCarInLoadout : 1; // 0x01D0 (0x0004)
[0x0000000000000000] [0x00000100]
struct FVector HandbrakeStartDir; // 0x01D4 (0x000C)
[0x0000000000000000]
struct FVector HandbrakeStartLocation; // 0x01E0 (0x000C)
class FString WastelandMapName; // 0x01F0 (0x0010)
[0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString AquaticMapName; // 0x0200 (0x0010)
[0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString UtopiaMapName; // 0x0210 (0x0010)
[0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString ArcMapName; // 0x0220 (0x0010)
[0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString TrainStationMapName; // 0x0230 (0x0010)
[0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> ProductNames_Body_MuscleCar; // 0x0240
(0x0010) [0x000000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> ProductNames_Body_SSFury; // 0x0250

```

```

(0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FName ProductName_Body_GearsCar; // 0x0260
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Import; // 0x0268 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Interceptor; // 0x0270
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Warthog; // 0x0278
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_WastelandTruck; // 0x0280
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Boost_Bubble; // 0x0288 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Boost_MusicalNotes; // 0x0290
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Hat_Halo; // 0x0298 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Octane; // 0x02A0 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Backfire; // 0x02A8 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Force; // 0x02B0 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Body_Vanquish; // 0x02B8
(0x0008) [0x0000000000000002] (CPF_Const)
TArray<struct FName> ProductNames_Body_Savage; // 0x02C0
(0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> ProductNames_Body_Spectacular; // 0x02D0
(0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> ProductNames_Body_Ruthless; // 0x02E0
(0x0010) [0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FName ProductName_Boost_Standard; // 0x02F0
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Boost_Ion; // 0x02F8 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName ProductName_Boost_Flamethrower; // 0x0300
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName ProductName_Boost_Thermal; // 0x0308
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName StatName_BoostTimeMinutes; // 0x0310
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName StatName_Certifiable; // 0x0318 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_DistanceDrivenKm; // 0x0320 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_ItemCollected; // 0x0328 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_LeftWingRightWing; // 0x0330 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_MadScientist; // 0x0338 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_MatchComplete; // 0x0340 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_WallTimeMinutes; // 0x0348 (0x0008)

```

```

[0x0000000000000002] (CPF_Const)
struct FName StatName_RocketGenocider; // 0x0350 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_BraveTheElements; // 0x0358 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_FullCourse; // 0x0360 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_BuckminsterX10; // 0x0368 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_ComingOnStrong; // 0x0370 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName StatName_PeoplePerson; // 0x0378 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName HockeyEventType; // 0x0380 (0x0008)
[0x0000000000000002] (CPF_Const)
class USaveData_TA* BaseSaveData; // 0x0388 (0x0008)
[0x0000000000000000]
class UAchievementSave_TA* SaveData; // 0x0390 (0x0008)
[0x0000000000000000]
class APRI_TA* LastPRI; // 0x0398 (0x0008)
[0x0000000000002000] (CPF_Transient)
class AGameInfo_Replay_TA* LastGame; // 0x03A0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class AGameEvent_TA* LastGameEvent; // 0x03A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ACar_TA* LastCar; // 0x03B0 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FGuid PendingTradeGuid; // 0x03B8 (0x0010)
[0x0000000000002000] (CPF_Transient)
class UProfileLoadoutSave_TA* LoadoutSave; // 0x03C8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UPlayerBannerSave_TA* PlayerBannerSave; // 0x03D0
(0x0008) [0x0000000000002000] (CPF_Transient)
class UPlayerAvatarBorderSave_TA* PlayerAvatarBorderSave; // 0x03D8
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate __EventNewMVPScore__Delegate; // 0x03DE0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventGoalAmountScored__Delegate; // 0x03EF8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventDrivenDistance__Delegate; // 0x04010
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventShotAmountBlocked__Delegate; // 0x04128
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventGameEventAmountWon__Delegate; // 0x04340
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventChampionshipAmountChanged__Delegate; // 0x04458
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventRandomItemAmountDropped__Delegate; // 0x04670
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventRegularSeasonAmountChanged__Delegate; // 0x04788
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventAchievementUnlocked__Delegate; // 0x049A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventStatIncremented__Delegate; // 0x04AB8

```

```
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSaveDataSet__Delegate;           // 0x04CD0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AchievementManager_TA");
}

return uClassPointer;
};

void __AchievementManager_TA__OnInit_0x2(class UShopsManager_TA* InShopManager);
void __AchievementManager_TA__OnInit_0x1(class USaveData_TA* _);
void __AchievementManager_TA__HandleShopPurchase_0x2(struct FOnlineProductData P);
bool __AchievementManager_TA__HandleShopPurchase_0x1(struct FOnlineProductData P);
void __AchievementManager_TA__ReconcileCollectionProgress_0x2(class UOnlineProduct_TA* P);
bool __AchievementManager_TA__ReconcileCollectionProgress_0x1(class UOnlineProduct_TA* P);
void __AchievementManager_TA__HandleProfileSet_0x3(class UPlayerVanitySave_TA* LoadedVanitySave);
void __AchievementManager_TA__HandleProfileSet_0x4(struct FProductInstanceId _);
void __AchievementManager_TA__HandleProfileSet_0x1(class UProfileLoadoutSave_TA* Save);
void __AchievementManager_TA__HandleProfileSet_0x2();
bool __AchievementManager_TA__CheckAllCarSlotsCustomized_0x1(class UProductSlot_TA* Slot);
bool __AchievementManager_TA__GetEquippedOnlineProducts_0x1(class UOnlineProduct_TA* P);
void GetAchievementProgress(int32_t AchievementId, uint8_t AchievementType, float& Out_Progress, float& Out_MaxProgress);
void GetClampedAchievementProgress(int32_t AchievementId, uint8_t AchievementType, float& Out_Progress, float& Out_MaxProgress);
void CheckBestOfTheBunch();
void CheckSquadGoals(class AGameEvent_Soccar_TA* GameEvent);
void CheckPeoplePerson(class AGameEvent_Soccar_TA* GameEvent);
void CheckNewChallenger(class AGameEvent_Soccar_TA* GameEvent);
void CheckTogetherIsBetter(class AGameEvent_Soccar_TA* GameEvent);
void CheckJoinTheClub(uint64_t ClubID);
void HandleClubChanged(class UOnlineClubManager_X* Manager, uint64_t ClubID);
void HandleSkillTierChanged(class APRI_TA* PRI);
void HandleTradeInComplete(class UGFxData_ProductTradeIn_TA* GFxData, TArray<class UOnlineProduct_TA*> Given, TArray<class UOnlineProduct_TA*> Received);
bool IsInCrossPlatformParty();
bool IsPlayingWithClubmates(class APRI_TA* PRI, class AGameEvent_Soccar_TA* GameEvent);
bool IsTournamentMatch();
bool IsOnlineMatch();
void CheckUnlockGoodTimes();
class UMapData_TA* GetMapData();
```

```
uint8_t GetWeatherVariant();
bool IsTurbulentWeather();
bool HasItemQualityEquipped(uint8_t Quality);
TArray<class UOnlineProduct_TA*> GetEquippedOnlineProducts();
void CheckUnlockStormTrooper();
void Save();
bool AddToIntList(int32_t Number, TArray<int32_t>& Out_IntList);
bool AddToNameList(struct FName NewName, TArray<struct FName>& Out_NameList);
bool AddToStringList(class FString NewName, TArray<class FString>& Out_StringList);
void ResetTemporaryVariables();
bool PlayerIsOnATeam();
bool NotifyKeyInput(int32_t ControllerId, struct FName Key, uint8_t EventType, float AmountDepressed, unsigned long bGamepad);
bool UpdateGoalsScoredAgainstBots(class AGameEvent_TA* GameEvent, int32_t MaxGoals, TArray<struct FName> RequiredProducts, float BotSkillLevel, int32_t& Out_UpdatedGoals);
void HandleScoredGoal(class APRI_TA* PRI, struct FVector BallHitLocation);
void HandlePlayerDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void SetHandbrakeStartData();
bool OtherTeamIsAllBots(class AGameEvent_TA* GameEvent);
bool ProductIsInAnyOpponentLoadout(class AGameEvent_TA* GameEvent, struct FName ProductName);
bool IsPlayingSnowDay(class AGameEvent_Soccar_TA* SoccarEvent);
bool QualifiesForDLC1Throwback();
bool ClassicCarlsInLoadout();
bool OrginalBattleCarlsInLoadout();
bool DLC1CarlsInLoadout();
void OnBreakoutPlatformDamaged(int32_t Count);
void HandleUsedAllMapVotes(class UMapPrefsSave_TA* MapPrefsSave);
void UpdateBraveTheElementsLevelsPlayed();
void CheckUnlockedBraveTheElements();
void CheckUnlockedDamageControl(class AGameEvent_Soccar_TA* GameEvent);
void CheckDLC0PowerSlide();
bool Handbrake180Completed();
int32_t GetTotalProductStats(TArray<struct FName>& ProductNames);
void CheckDLC0WheelDriveDistance();
void CheckDLC0BoostTime();
void HandleStatsCommitted(class APRI_TA* PRI);
bool AtleastOneProductInLoadout(TArray<struct FName> ProductNames);
bool ProductIsInLoadout(struct FName ProductName);
bool IsWearingDecal();
bool IsWearingHat();
bool PlayingMapOrVariant(struct FName MapName);
bool PlayingMap(class FString MapName);
void CheckAllCarSlotsCustomized(int32_t ProductID);
void HandleReplayStarted(class AGameInfo_Replay_TA* Game);
void CheckProfileCustomized();
void HandleVanityChanged();
void HandlePlayerTitleChanged();
void OnProductEquipped(int32_t ProductID);
void HandleProductEquipped(int32_t ProductID);
void ListenForRegisteredVoter();
void HandleProfileSet(class ULocalPlayer_TA* LocalPlayer);
bool AllSeasonGamesPlayedWithDLC0Cars(class AGameEvent_Season_TA* GameEvent);
void HandleWonChampionship(class AGameEvent_Season_TA* GameEvent);
```

```
void UpdateDriveTime();
void CheckBoosting(float DeltaTime);
void CheckWallDriveTime(float DeltaTime);
void eventTick(float DeltaTime);
void HandleBoostAmountChanged(class ACarComponent_Boost_TA* Boost);
void CheckTrainingModesPlayed();
void CheckGameProgress();
void UpdateTrainingModesPlayed(class AGameEvent_TA* GameEvent);
bool PlayedAllLevels(TArray<struct FName>& RequiredLevels);
void UpdateLevelsPlayed();
void IncrementComingOnStrong();
void HandleStatEvent(class APRI_TA* PRI, class UStatEvent_TA* StatEvent, int32_t Count);
void CheckTrifecta();
void CheckUnlockedCars();
void HandleCertifiedStatRankedUp(class UProductAttribute_Certified_TA* CertifiedAttribute);
void HandleNewOnlineProduct(class USaveData_TA* Data, class UOnlineProduct_TA*
OnlineProduct, class FString Message);
void CheckCollectedProducts();
void BroadcastRandomItemsDropped();
void IncrementCollectedProducts(int32_t Count);
void ReconcileCollectionProgress();
void HandleShopPurchase(TArray<struct FOnlineProductData> PurchasedProducts,
TArray<struct FCurrency> PurchasedCurrencies);
bool IsPlayingWithAFriend(class APRI_TA* PlayerPRI);
bool PlayedAnyCar(TArray<struct FName> CarsToCheck);
bool PlayedCar(struct FName CarToCheck);
bool PlayedAllCars(TArray<struct FName> CarsToCheck);
void UpdateCarsPlayed();
void CheckPlayedWithFriends(class AGameEvent_Soccar_TA* GameEvent);
void HandleMatchEnded(class AGameEvent_Soccar_TA* GameEvent);
void CheckBotTeamsPlayed(class AGameEvent_Soccar_TA* GameEvent);
void UpdateBotTeamsPlayed(class AGameEvent_Soccar_TA* GameEvent);
void NotifyGameEventWon();
void HandleMatchWinnerSet(class AGameEvent_Soccar_TA* GameEvent);
void HandlePlayerCarSet(class APRI_TA* PRI);
void HandleNewPlayerAdded(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* GameEvent);
bool GameEventHasPlayersOnOtherTeam(class AGameEvent_TA* GameEvent);
bool IsTiedOrTrailingOtherTeam(class AGameEvent_Soccar_TA* GameEvent);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* GameEvent);
void HandleCarAddComponent(class UObject* O);
void SetupDriftKingForUnlock();
void InitCar();
void HandlePickupActivated(class ACar_TA* InCar, class ASpecialPickup_TA* ActivatedPickup);
void CheckAllRumbleActivations(class ASpecialPickup_TA* ActivatedPickup);
struct FName GetRumbleItemName(class ASpecialPickup_TA* Pickup);
void HandleCarSet(class APRI_TA* InPRI);
void HandleTeamChanged(class APRI_X* InPRI);
void HandleGameEventChanged(class APRI_TA* PRI);
void UpdatePRI(class APRI_TA* PRI);
void HandleReceivedPRI(class APlayerController_X* InPC);
void OnInit();
void SetAchievementSave(class UAchievementSave_TA* Save);
void SetBaseSaveData(class USaveData_TA* MySaveData);
```

```

void HandleBaseSaveDataLoaded(class USaveGameManager_TA* Manager, class
USaveData_TA* MySaveData, class UError* Error);
void HandlePCDestroyed(class APlayerController_X* InPC);
void HandlePlayerReceived(class APlayerControllerBase_TA* InPC);
void Init(class APlayerControllerBase_TA* InPC);
void SyncPS4Achievements(TArray<unsigned long> SyncedAchievements);
void CheckPS4AchievementsSynced();
struct FUniqueNetId GetUniqueNetId();
void SyncSteamAchievements(int32_t TitleId);
void OnFirstExecution();
void FlushOnlineStats();
void IncrementStat(int32_t AchievementId, uint8_t AchievementType, int32_t Progress);
static bool AchievementTypeExistsOnPlatform(uint8_t AchievementType);
static int32_t GetAchievementTypeMax(uint8_t AchievementType);
int32_t GetAchievementIconID(int32_t InAchievementID, uint8_t AchievementType);
void GetAchievementIDAndType(int32_t GlobalAchievementID, int32_t& Out_AchievementID,
uint8_t& Out_AchievementType);
static int32_t GetGlobalAchievementID(int32_t InAchievementID, uint8_t AchievementType);
bool IsUnlocked(int32_t AchievementId, uint8_t AchievementType, unsigned long
bIgnoreAuthority);
void SetAchievementUnlocked(int32_t AchievementId, unsigned long bUnlocked);
void UnlockAchievement(int32_t AchievementId, uint8_t AchievementType, unsigned long
bIgnoreAuthority);
void OnProgressionChanged(int32_t AchievementId, uint8_t AchievementType);
void EventSaveDataSet(class UAchievementManager_TA* Manager);
void EventStatIncremented(class UAchievementManager_TA* Manager, int32_t AchievementId,
uint8_t AchievementType);
void EventAchievementUnlocked(class UAchievementManager_TA* Manager, int32_t
AchievementId, uint8_t AchievementType);
void EventRegularSeasonAmountChanged(class UAchievementManager_TA* Manager, int32_t
Amount);
void EventRandomItemAmountDropped(class UAchievementManager_TA* Manager, int32_t
Amount);
void EventChampionshipAmountChanged(class UAchievementManager_TA* Manager, int32_t
Wins);
void EventGameEventAmountWon(class UAchievementManager_TA* Manager, int32_t
GamesWon);
void EventShotAmountBlocked(class UAchievementManager_TA* Manager, int32_t bLocked);
void EventDrivenDistance(class UAchievementManager_TA* Manager, float Distance);
void EventGoalAmountScored(class UAchievementManager_TA* Manager, int32_t Score);
void EventNewMVPScore(class UAchievementManager_TA* Manager, int32_t Score);
};

// Class TAGame.__AchievementManager_TA__HasItemQualityEquipped_0x1
// 0x0001 (0x0060 - 0x0061)
class U__AchievementManager_TA__HasItemQualityEquipped_0x1 : public UObject
{
public:
    uint8_t           Quality;          // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._AchievementManager_TA__HasItemQualityEquipped_0x1");
}

return uClassPointer;
};

bool __AchievementManager_TA__HasItemQualityEquipped_0x1(class UOnlineProduct_TA* P);
};

// Class TAGame._AchievementManager_TA__IsPlayingWithClubmates_0x1
// 0x0008 (0x0060 - 0x0068)
class U_AchievementManager_TA__IsPlayingWithClubmates_0x1 : public UObject
{
public:
class APRI_TA*          PRI;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._AchievementManager_TA__IsPlayingWithClubmates_0x1");
}

return uClassPointer;
};

bool __AchievementManager_TA__IsPlayingWithClubmates_0x1(class APRI_TA* P);
};

// Class TAGame._ArenaSoundConfig_TA__GetNextEvent_0x1
// 0x0010 (0x0060 - 0x0070)
class U_ArenaSoundConfig_TA__GetNextEvent_0x1 : public UObject
{
public:
struct FName             MapName;        // 0x0060 (0x0008)
[0x0000000000000000]
uint64_t                 CurrentTime;    // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._ArenaSoundConfig_TA__GetNextEvent_0x1");
}

return uClassPointer;
};

bool __ArenaSoundConfig_TA__GetNextEvent_0x1(class UArenaSoundEvent_TA* E);
};

// Class TAGame.ArenaSoundEvent_TA
// 0x0020 (0x0060 - 0x0080)
class UArenaSoundEvent_TA : public UObject
{
public:
struct FName MapName; // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue* InGameMusicCue; // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
uint64_t StartTime; // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)
uint64_t EndTime; // 0x0078 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ArenaSoundEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.ArenaSoundConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UArenaSoundConfig_TA : public UOnlineConfig_X
{
public:
TArray<class UArenaSoundEvent_TA*> Events; // 0x0078 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ArenaSoundConfig_TA");
}

return uClassPointer;
};

bool __ArenaSoundConfig_TA__GetNextEvent_0x3(class UArenaSoundEvent_TA* E);
int32_t __ArenaSoundConfig_TA__GetNextEvent_0x2(class UArenaSoundEvent_TA* L, class
UArenaSoundEvent_TA* R);
class UArenaSoundEvent_TA* GetNextEvent(struct FName MapName, uint64_t CurrentTime);
};

// Class TAGame.__ArenaSoundManager_TA__HandleMatchEnded_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ArenaSoundManager_TA__HandleMatchEnded_0x1 : public UObject
{
public:
class AGameEvent_Soccar_TA*           InGameEvent;          // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ArenaSoundManager_TA__HandleMatchEnded_0x1");
}

return uClassPointer;
};

bool __ArenaSoundManager_TA__HandleMatchEnded_0x1(class APRI_TA* PRI);
};

// Class TAGame.GameEvent_Soccar_TA
// 0x0500 (0x07A0 - 0x0CA0)
class AGameEvent_Soccar_TA : public AGameEvent_Team_TA
{
public:
class ACar_TA*           TestCarArchetype;        // 0x07A0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class ABall_TA*           BallArchetype;          // 0x07A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AGoalIndicator_TA*   GoalIndicatorArchetype; // 0x07B0
(0x0008) [0x0000000000000000]
class AActor*              BallSpawnPoint;         // 0x07B8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AStatFactory_TA*     StatFactoryArchetype; // 0x07C0 (0x0008)
[0x0000000000000001] (CPF_Edit)

```

```

struct FName InitialEventState; // 0x07C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t SeriesLength; // 0x07D0 (0x0004)
[0x0000000000000021] (CPF_Edit | CPF_Net)
int32_t GameTime; // 0x07D4 (0x0004)
[0x0000000000000021] (CPF_Edit | CPF_Net)
int32_t WarmupTime; // 0x07D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t MaxScore; // 0x07DC (0x0004)
[0x0000000000000021] (CPF_Edit | CPF_Net)
int32_t AutoBalanceDifference; // 0x07E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UOvertimeComponent_TA* OvertimeComponent; // 0x07E8
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FInterpCurveFloat ScoreSlomoCurve; // 0x07F0 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float ScoreSlomoTime; // 0x0808 (0x0004)
[0x0000000000002000] (CPF_Transient)
float GameTimeRemaining; // 0x080C (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t SecondsRemaining; // 0x0810 (0x0004)
[0x0000008100002020] (CPF_Net | CPF_Transient)
int32_t WaitTimeRemaining; // 0x0814 (0x0004)
[0x0008000000002020] (CPF_Net | CPF_Transient)
float TotalGameTimePlayed; // 0x0818 (0x0004)
[0x0000000000002000] (CPF_Transient)
float OvertimeTimePlayed; // 0x081C (0x0004)
[0x0000008000002000] (CPF_Transient)
class UMessage_TA* GoalScoredMessage; // 0x0820 (0x0008)
[0x0000000000000000]
class UMessage_TA* OvertimeMessage; // 0x0828 (0x0008)
[0x0000000000000000]
class UMessage_TA* TwoMinRemainingMessage; // 0x0830
(0x0008) [0x0000000000000000]
class UMessage_TA* OneMinRemainingMessage; // 0x0838
(0x0008) [0x0000000000000000]
class UMessage_TA* ThirtySecondsRemainingMessage; // 0x0840
(0x0008) [0x0000000000000000]
class UMessage_TA* StopMessage; // 0x0848 (0x0008)
[0x0000000000000000]
class UMessage_TA* GamePausedMessage; // 0x0850
(0x0008) [0x0000000000000000]
class UMessage_TA* GameUnPausedMessage; // 0x0858
(0x0008) [0x0000000000000000]
class UMessage_TA* OutOfBoundsMessage; // 0x0860
(0x0008) [0x0000000000000000]
unsigned long bRoundActive : 1; // 0x0868 (0x0004)
[0x000008000002000] [0x00000001] (CPF_Transient)
unsigned long bPlayReplays : 1; // 0x0868 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bBallHasBeenHit : 1; // 0x0868 (0x0004)
[0x0000008100002020] [0x00000004] (CPF_Net | CPF_Transient)
unsigned long bOverTime : 1; // 0x0868 (0x0004)

```

[0x0000004100002020] [0x00000008] (CPF\_Net | CPF\_Transient)  
unsigned long bUnlimitedTime : 1; // 0x0868 (0x0004)  
[0x0000000000000020] [0x00000010] (CPF\_Net)  
unsigned long bNoContest : 1; // 0x0868 (0x0004)  
[0x0000004000002020] [0x00000020] (CPF\_Net | CPF\_Transient)  
unsigned long bDisableGoalDelay : 1; // 0x0868 (0x0004)  
[0x0000004000002000] [0x00000040] (CPF\_Transient)  
unsigned long bShowNoScorerGoalMessage : 1; // 0x0868  
(0x0004) [0x0000000000000002] [0x00000080] (CPF\_Const)  
unsigned long bMatchEnded : 1; // 0x0868 (0x0004)  
[0x0000004100002020] [0x00000100] (CPF\_Net | CPF\_Transient)  
unsigned long bShowIntroScene : 1; // 0x0868 (0x0004)  
[0x0008000000000020] [0x00000200] (CPF\_Net)  
unsigned long bClubMatch : 1; // 0x0868 (0x0004)  
[0x0009000100000020] [0x00000400] (CPF\_Net)  
unsigned long bCanDropOnlineRewards : 1; // 0x0868 (0x0004)  
[0x0008000000000020] [0x00000800] (CPF\_Net)  
unsigned long bAllowHonorDuels : 1; // 0x0868 (0x0004)  
[0x0001000100002020] [0x00001000] (CPF\_Net | CPF\_Transient)  
int32\_t NextSpawnIndex; // 0x086C (0x0004)  
[0x0000000000002000] (CPF\_Transient)  
class AReplayDirector\_TA\* ReplayDirectorArchetype; // 0x0870  
(0x0008) [0x0000000000000001] (CPF\_Edit)  
class AReplayDirector\_TA\* ReplayDirector; // 0x0878 (0x0008)  
[0x0000008100002020] (CPF\_Net | CPF\_Transient)  
TArray<class ABall\_TA\*> GameBalls; // 0x0880 (0x0010)  
[0x0000008000402000] (CPF\_Transient | CPF\_NeedCtorLink)  
int32\_t TotalGameBalls; // 0x0890 (0x0004)  
[0x0000004000000001] (CPF\_Edit)  
float PostGoalTime; // 0x0894 (0x0004)  
[0x0000000000000001] (CPF\_Edit)  
class AStatFactory\_TA\* StatFactory; // 0x0898 (0x0008)  
[0x0000004000002000] (CPF\_Transient)  
TArray<class UGoal\_TA\*> Goals; // 0x08A0 (0x0010)  
[0x0000008004482008] (CPF\_ExportObject | CPF\_Transient | CPF\_Component |  
CPF\_NeedCtorLink | CPF\_EditInline)  
int32\_t SecondsRemainingCountdown; // 0x08B0 (0x0004)  
[0x0000000000000001] (CPF\_Edit)  
class UMessage\_TA\* SecondsRemainingMessage; // 0x08B8  
(0x0008) [0x0000000000000001] (CPF\_Edit)  
struct FVector FieldCenter; // 0x08C0 (0x000C)  
[0x0000004000002000] (CPF\_Transient)  
class ATeam\_TA\* GameWinner; // 0x08D0 (0x0008)  
[0x0000004100002020] (CPF\_Net | CPF\_Transient)  
class ATeam\_TA\* MatchWinner; // 0x08D8 (0x0008)  
[0x0000008100002020] (CPF\_Net | CPF\_Transient)  
uint8\_t TieBreakDecision; // 0x08E0 (0x0001)  
[0x0008000000002020] (CPF\_Net | CPF\_Transient)  
uint8\_t ReplicatedScoredOnTeam; // 0x08E1 (0x0001)  
[0x0000004100002020] (CPF\_Net | CPF\_Transient)  
uint8\_t ReplicatedServerPerformanceState; // 0x08E2 (0x0001)  
[0x0008000000000020] (CPF\_Net)  
class APRI\_TA\* MVP; // 0x08E8 (0x0008)  
[0x0000004000002020] (CPF\_Net | CPF\_Transient)

```

class APRI_TA*           FastestGoalPlayer;          // 0x08F0 (0x0008)
[0x0000004000002000] (CPF_Transient)
class APRI_TA*           SlowestGoalPlayer;         // 0x08F8 (0x0008)
[0x0000004000002000] (CPF_Transient)
class APRI_TA*           FurthestGoalPlayer;        // 0x0900 (0x0008)
[0x0000004000002000] (CPF_Transient)
float                   FastestGoalSpeed;          // 0x0908 (0x0004)
[0x0000004000002000] (CPF_Transient)
float                   SlowestGoalSpeed;          // 0x090C (0x0004)
[0x0000004000002000] (CPF_Transient)
float                   FurthestGoal;              // 0x0910 (0x0004)
[0x0000004000002000] (CPF_Transient)
class APRI_TA*           ScoringPlayer;            // 0x0918 (0x0008)
[0x0000008000002000] (CPF_Transient)
int32_t                RoundNum;                  // 0x0920 (0x0004)
[0x0000004000002020] (CPF_Net | CPF_Transient)
float                   AssistMaxTime;             // 0x0924 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                   BallHasBeenHitStartDelay; // 0x0928 (0x0004)
[0x0000000000000000]
float                   BallRespawnTime;           // 0x092C (0x0004)
[0x0000004000002000] (CPF_Transient)
class UGameEvent_Soccar_SubRules_TA*   DefaultSubRules; // 0x0930
(0x0008) [0x0000000000000001] (CPF_Edit)
class UGameEvent_Soccar_SubRules_TA*   SubRulesArchetype; // 0x0938
(0x0008) [0x0000000100002020] (CPF_Net | CPF_Transient)
class UGameEvent_Soccar_SubRules_TA*   SubRules;        // 0x0940
(0x0008) [0x0000000000002000] (CPF_Transient)
class USpawnPointCluster_TA*         SpawnPointsPodiumAnim; // 0x0948
(0x0008) [0x0000000000000000]
class USpawnPointCluster_TA*         SpawnPointsPodium;    // 0x0950
(0x0008) [0x0000000000000000]
float                   PodiumDelay;              // 0x0958 (0x0004)
[0x0000000000000000]
float                   PodiumTime;               // 0x095C (0x0004)
[0x0000000000000000]
TArray<class USpawnPointCluster_TA*>   SpawnPointLobbyTeams; // 0x0960 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<class UStatCategory_TA*>       StatCategories; // 0x0970
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class ABall_Trajectory_TA*           BallTrajectory; // 0x0980 (0x0008)
[0x0000000000000000]
class ABall_Trajectory_TA*           BallTrajectoryArchetype; // 0x0988
(0x0008) [0x0000000000000000]
class UCountdownObject_TA*          UnpauseCountdown; // 0x0990
(0x0008) [0x0000004000002000] (CPF_Transient)
class APlayerController_TA*         Pauser;        // 0x0998 (0x0008)
[0x0000004000002000] (CPF_Transient)
TArray<struct FOnlinePlayerMatchData> PlayerMatchData; // 0x09A0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UEndRoundComponent_TA*        EndRoundComponentArchetype; // 0x09B0 (0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UEndRoundComponent_TA*        EndRoundComponent; // 0x09B8

```

```

(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UMatchBroadcastComponent_TA*           MatchBroadcast;          // 0x09C0
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UStatEvent_TA*                        ReplicatedStatEvent;    // 0x09C8 (0x0008)
[0x0000008100002020] (CPF_Net | CPF_Transient)
TArray<class UStatEvent_TA*>               ImportantStatEvents;   // 0x09D0
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UGameObserver_TA*                     GameObserver;          // 0x09E0 (0x0008)
[0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UFpsBucketRecorder_TA*                ActiveFpsRecorder;     // 0x09E8
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UFpsBucketRecorder_TA*                InactiveFpsRecorder;   // 0x09F0
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UServerPerformanceTracker_TA*        ServerPerformanceTracker; // 0x09F8 (0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UMatchSeries_TA*                     MatchSeries;           // 0x0A00 (0x0008)
[0x0000800000000001] (CPF_Edit)
class UCrowdSoundManagerBase_TA*           CrowdSound;           // 0x0A08
(0x0008) [0x00100004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
float                           LobbyTagOffsetZ;      // 0x0A10 (0x0004)
[0x0000000000000002] (CPF_Const)
float                           PodiumTagOffsetZ;     // 0x0A14 (0x0004)
[0x0000000000000002] (CPF_Const)
float                           BallSpacing;          // 0x0A18 (0x0004)
[0x0000000000000002] (CPF_Const)
float                           PreMatchPlayersReadyStartDelay; // 0x0A1C (0x0004)
[0x0000000000000002] (CPF_Const)
TArray<class APRI_TA*>                  RemovedPRIs;          // 0x0A20 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate                 _EventActiveRoundChanged__Delegate; // 0x0A30
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventStartNewRound__Delegate; // 0x0A48
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventBallAdded__Delegate;  // 0x0A60
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventBallRemoved__Delegate; // 0x0A78
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventFirstBallHit__Delegate; // 0x0A90
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventGoalScored__Delegate; // 0x0AA8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventGameTimeUpdated__Delegate; // 0x0AC0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventOvertimeUpdated__Delegate; // 0x0AD8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventGameEnded__Delegate;  // 0x0AF0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventMatchEnded__Delegate; // 0x0B08
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventEndGameCountDown__Delegate; // 0x0B20

```

```

(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventReplayDirectorSet__Delegate;      // 0x0B38
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventGameWinnerSet__Delegate;        // 0x0B50
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventMatchWinnerSet__Delegate;       // 0x0B68
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPlayerScored__Delegate;        // 0x0B80
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventReplicatedGoalScored__Delegate; // 0x0B98
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventStatFactoryInitialized__Delegate; // 0xBB0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventReplicatedStatEvent__Delegate;   // 0xBC8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventMaxScoreChanged__Delegate;      // 0xBE0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventReplayFinished__Delegate;       // 0xBF8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __WaitTimeRemaining__ChangeNotify;     // 0xC10
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bShowIntroScene__ChangeNotify;      // 0xC28
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bClubMatch__ChangeNotify;          // 0xC40
(0x0018) [0x00100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bCanDropOnlineRewards__ChangeNotify; // 0xC58
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __TieBreakDecision__ChangeNotify;    // 0xC70
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __ReplicatedServerPerformanceState__ChangeNotify;// 0xC88
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Soccar_TA");
}

return uClassPointer;
};

void CheckStart();
int32_t GetPlayerCarCount();
bool GetShouldStart();
void ReplicateSkillTiers();
bool StartIntroScene();
void StopIntroScene();
void DisableGameBall(class ABall_TA* Ball, unsigned long bDisable);
void DisableGameBalls(unsigned long bDisable);
void StartMatch();

```

```
bool CanSpawnBots();
void HandlePlayerReadyUp(class APRI_TA* PRI);
void EnableFreestyle(class APRI_TA* PRI);
void StartRound();
void EndRound();
void SetBallEventListeners(class ABall_TA* Ball, unsigned long bListen);
bool CanAwardPoints();
void HandleCarTouch(class ABall_TA* Ball, class ACar_TA* HitCar, uint8_t HitType);
void SetBallHasBeenHit();
int32_t DetermineScoreTouchIndex(class ABall_TA* Ball, class UGoal_TA* Goal);
int32_t DetermineAssistTouchIndex(class ABall_TA* Ball, int32_t ScoreIdx);
void UpdateTotalGameTimePlayed(float DeltaTime);
float GetDesiredDeltaTime(float DeltaTime);
void UpdateGameTime(float DeltaTime);
void WaitForEndRound();
bool CanUpdateGameTime();
void AddControllerToRespawnList(class AController* InController);
void StartReplay();
void HandleReplayFinished(class AReplayDirector_TA* InReplay);
void GotoPodiumSpotlight();
int32_t MemberTitleSort(struct FMemberTitleStat A, struct FMemberTitleStat B);
void UpdateSpotlight();
void InitPodiumCars();
TArray<class APRI_TA*> DeterminePodiumCars();
bool CanEnableCarPodiumMovement();
void FinishEvent();
void HandleNameplateSet(class UNameplateMeshComponent_TA*
NameplateMeshComponent);
bool __GameEvent_Soccar_TA__InitClubMatch_0x3(struct FReservationData P);
bool __GameEvent_Soccar_TA__InitClubMatch_0x1(struct FReservationData P);
bool __GameEvent_Soccar_TA__UpdateTeamScores_0x1(class ATeam_TA* T);
void __GameEvent_Soccar_TA__SubmitMatchComplete_0x1(class APRI_TA* PRI);
void __GameEvent_Soccar_TA__CheckStart_0x1(class ATeam_TA* T);
bool __GameEvent_Soccar_TA__CanUpdateGameTime_0x1(class ATeam_TA* Team);
bool __GameEvent_Soccar_TA__CanSpawnBots_0x1(class ATeam_TA* Team);
void __GameEvent_Soccar_TA__EndState_0x1(class ATeam_TA* T);
struct FUniqueNetId __GameEvent_Soccar_TA__CheckJoinInProgress_0x1(class APRI_TA* PRI);
int32_t __GameEvent_Soccar_TA__GetIsSeriesComplete_0x1(class ATeam_Soccar_TA* T);
void __ReplicatedServerPerformanceState__ChangeNotifyFunc();
void __TieBreakDecision__ChangeNotifyFunc();
void __bCanDropOnlineRewards__ChangeNotifyFunc();
void __bClubMatch__ChangeNotifyFunc();
void __bShowIntroScene__ChangeNotifyFunc();
void __WaitTimeRemaining__ChangeNotifyFunc();
void OnMaxScoreChanged();
class UGoal_TA* GetGoalByTeamIndex(int32_t TeamIndex);
bool GetIsSeriesComplete();
void CheckJoinInProgress(class APRI_TA* PRI);
bool AllowDynamicCrowd();
void HandleTrajectoryEnabledChanged(class ABall_TA* InBall);
void DestroyBallTrajectory();
void AddBallTrajectory(class ABall_TA* InBall);
static class FString GetInGamePresence(class FString PlaylistName, class FString MapName,
unsigned long bLocalize);
```

```
void GetPresenceStatus(unsigned long bLocalize, class FString& Out_PresenceString, class
FString& Out_GameDataString);
bool ShowScorerGoalMessage();
bool CanUseBallCam();
bool RestartPlayer(class AController* NewPlayer);
bool DisableStatXP();
void SetDisableGoalDelay(unsigned long bInDisableGoalDelay, float InBallRespawnTime);
void ForceMatchStart();
void RemoveLocalPlayer(class APlayerController_TA* Player);
void AddLocalPlayer(class APlayerController_TA* Player);
void DestroyGoalIndicators(class APlayerController_TA* Player);
class AGoalIndicator_TA* SpawnGoalIndicator(class APlayerController_TA* Player, class
UGoal_TA* Goal, class AActor* OrientationActor);
void CreateGoalIndicators(class APlayerController_TA* Player);
void HandleMatchCompleteDataReceived(class URPC_MatchComplete_TA* RPC);
void BeginHighlightsReplay();
void SetGoalsEnabled(unsigned long bEnabled);
void OnPlayerOutOfBounds(class APlayerController_TA* PC);
int32_t GetGoalScorePoints(class ABall_TA* Ball, class APRI_TA* ScorerPRI, class UGoal_TA*
Goal);
bool ShowSeasonIntroScene();
bool ShouldCountUp();
bool ShouldAllowVoteToForfeit();
bool ShouldHaveLeaveMatchPenalty();
void RemovePlayer(class AController* Player);
void HandleUnpauseTimer(class UCountdownObject_TA* CountdownObject, int32_t
CountValue);
void SetPaused(class APlayerController_TA* InPauser, unsigned long bInPaused);
bool ShouldCountdownResumeFromPause();
void SetScoreAndTime(class APlayerController_TA* PC, int32_t NewScoreTeam0, int32_t
NewScoreTeam1, int32_t InGameTimeRemaining, unsigned long bInOvertime, unsigned long
bRestartRound);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void SaveLocalPlayerStats();
bool ShouldPlayReplay();
bool ShouldRecordReplay();
void OnBallHasBeenHit();
class ABall_TA* SpawnBall(struct FVector SpawnLoc, unsigned long bWake, unsigned long
bSpawnCannon, class FString BallArch);
struct FName GetScoreStatus(uint8_t TeamNum);
int32_t GetTotalScore();
struct FName GetDesiredSoundState();
void HandleCarSet(class APRI_TA* InPRI);
void RemovePRI(class APRI_TA* PRI);
void AddPRI(class APRI_TA* PRI);
void AddPlayerToTeam(class ATeam_TA* NewTeam, class AController* NewPlayer);
class ATeam_TA* PickTeamForSeries(class AController* C);
class ATeam_TA* PickTeam(class AController* C);
bool ShouldAutoSelectTeam(class AController* C);
void OnMatchWinnerSet();
void OnGameWinnerSet();
int32_t ScoreboardSort(class APRI_TA* A, class APRI_TA* B);
class APRI_TA* GetMVP(class ATeam_TA* WinningTeam);
int32_t MVPSort(class APRI_TA* A, class APRI_TA* B);
```

```
void HandleHitGoal(class ABall_TA* Ball, class UGoal_TA* Goal);
void ClearReplicatedScoredOnTeam();
void TriggerScoreChangedEvent();
void HandleScoreUpdated(class ATeam_TA* Team);
void OnAllTeamsCreated();
void TriggerGoalScoreEvent(int32_t TeamScoredOn, class ACar_TA* Scorer);
void SetTotalGameBalls(int32_t TotalBalls);
void RecordRecentPlayers();
void UpdateStats();
void AddStatData(struct FName StatId, struct FUniqueNetId PlayerID, int32_t Value, TArray<struct FUploadStatDataSet>& DataSet);
void NotifyKismetOfCurrentTime();
bool EnoughTimePassedToForfeit();
bool ShowSecondsRemainingCountdown();
void OnGameTimeUpdated();
void OnOvertimeUpdated();
void ForceOvertime();
void StartOvertime();
bool OnMyHalf(struct FVector TestLocation, uint8_t TeamNum);
class ATeam_TA* GetWinningTeam();
void SetMatchWinner(class ATeam_TA* Team, uint8_t InTieBreakDecision);
void OnBallSpawned(class ABall_TA* NewBall);
void ResetBalls();
bool GetBallSpawnOrientation(struct FVector& out_Location, struct FRotator& out_Rotation);
float CalcuateBallRadius();
void FreezePawns();
void DestroyBalls();
void RemoveGameBall(class ABall_TA* Ball);
void AddGameBall(class ABall_TA* Ball);
void StartNewRound();
void CheckForAutoBalance();
bool HasMatchWinner();
bool HasWinner();
void CheckForForfeit(class AController* RemovedPlayer, class ATeam_TA* Team);
void SubmitMatch();
bool IsMatchWinner(class APRI_TA* PRI);
void CommitPlayerMatchData(class APRI_TA* PRI, unsigned long bCompletedMatch);
void SubmitMatchComplete();
void OnMatchEnded();
void NotifyOnGameEnded(struct FScriptDelegate Callback);
struct FName GetEndMatchState();
void EndGame();
void UpdateTeamScores();
void StartNewGame();
void ResetToDefaultTime();
void ResetGame();
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct FRotator& out_Rotation);
void ClearReplicatedStatEvent();
void HandleGaveStat(class AStatFactory_TA* Factory, class APRI_TA* ToPRI, class UStatEvent_TA* StatEvent, class ABall_TA* Ball, int32_t BallHitIndex, class APRI_TA* Victim, class UGoal_TA* Goal);
void HandlePlayerSkillUpdated(struct FUniqueNetId PlayerID, struct FUpdatedPlayerSkillRating Rating);
```

```

void ReportFps(class UFpsBucketRecorder_TA* Recorder, unsigned long bIsActiveFpsReport);
void eventDestroyed();
void InitCrowdManager();
void InitField();
void InitGameObserver();
void OnInit();
void InitMutators();
void OnClubMatch();
void InitClubMatch();
bool CanInitClubMatch();
void SetSeriesScore();
void AssignCustomTeamSettings();
void InitGame(class FString Options);
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventReplayFinished(class AGameEvent_Soccar_TA* GameEvent);
void EventMaxScoreChanged(class AGameEvent_Soccar_TA* GameEvent);
void EventReplicatedStatEvent(class AGameEvent_Soccar_TA* GameEvent, class UStatEvent_TA* StatEvent);
void EventStatFactoryInitialized(class AGameEvent_Soccar_TA* GameEvent);
void EventReplicatedGoalScored(class AGameEvent_Soccar_TA* GameEvent, uint8_t ScoredOnTeam);
void EventPlayerScored(class AGameEvent_Soccar_TA* GameEvent, class APRI_TA* Scorer);
void EventMatchWinnerSet(class AGameEvent_Soccar_TA* GameEvent);
void EventGameWinnerSet(class AGameEvent_Soccar_TA* GameEvent);
void EventReplayDirectorSet(class AGameEvent_Soccar_TA* GameEvent);
void EventEndGameCountDown(class AGameEvent_Soccar_TA* GameEvent, int32_t TimeRemaining);
void EventMatchEnded(class AGameEvent_Soccar_TA* GameEvent);
void EventGameEnded(class AGameEvent_Soccar_TA* GameEvent);
void EventOvertimeUpdated(class AGameEvent_Soccar_TA* GameEvent);
void EventGameTimeUpdated(class AGameEvent_Soccar_TA* GameEvent);
void EventGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
void EventFirstBallHit(class AGameEvent_Soccar_TA* GameEvent);
void EventBallRemoved(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball);
void EventBallAdded(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball);
void EventStartNewRound(class AGameEvent_Soccar_TA* GameEvent);
void EventActiveRoundChanged(class AGameEvent_Soccar_TA* GameEvent);
};

// Class TAGame.ArenaSoundManager_TA
// 0x00BC (0x00A4 - 0x0160)
class UArenaSoundManager_TA : public UActorComponent_X
{
public:
    class UAkSoundCue*           DefaultInGameMusicCue;          // 0x00A8
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           TrainingRoundEndCue;           // 0x00B0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           ArenaMusicReplayStopCue;        // 0x00B8
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class UArenaSoundSettings_TA* Settings;                      // 0x00C0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
}

```

```

class AArenaSoundPlayer_TA*           SoundPlayer;          // 0x00C8 (0x0008)
[0x0000000000002000] (CPF_Transient)
unsigned long                      bMatchBegan : 1;      // 0x00D0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
class AGameEvent_Soccar_TA*         SoccarGame;          // 0x00D8
(0x0008) [0x0000000000002000] (CPF_Transient)
class UArenaReflectionsManager_TA*   ReflectionsManager; // 0x00E0
(0x0008) [0x0000000000000000]
class UAkSoundCue*                 GoalScoreStinger;    // 0x00E8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UProductAsset_MusicStingers_TA* ActiveStinger;     // 0x00F0
(0x0008) [0x0000004000002000] (CPF_Transient)
class UArenaSoundConfig_TA*         Config;              // 0x00F8 (0x0008)
[0x0000800000000000]
class UStreamerSafeConfig_TA*       StreamerSafeConfig; // 0x0100
(0x0008) [0x0018000000000000]
class UArenaSoundEvent_TA*          NextTimedEvent;     // 0x0108
(0x0008) [0x0000000000000000]
int32_t                           TimedEventPlayingID; // 0x0110 (0x0004)
[0x0000000000000000]
struct FScriptDelegate             __EventReflectionsManagerSet__Delegate; // 0x0118
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __EventActiveStingerUpdated__Delegate; // 0x0130
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __EventAnthemSoundStateUpdated__Delegate; // 0x0148
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ArenaSoundManager_TA");
}

return uClassPointer;
};

void __ArenaSoundManager_TA__StreamerSafeConfigChanged_0x1();
void __ArenaSoundManager_TA__RegisterGameEventCallbacks_0x1(class AGameEvent_TA* G,
class ACar_TA* C);
void ClearBallExplodeState();
void HandleBallExploded(class ABall_TA* Ball);
void HandlePostTimeSkip(class UReplay_TA* Replay);
bool IsFreeplay();
bool IsTraining();
bool IsReplay();
void HandleStatTickerMessage(class APRI_TA* Receiver, class APRI_TA* Victim, class
UStatEvent_TA* StatEvent);
void HandlePlayerScored(class AGameEvent_Soccar_TA* GameEvent, class APRI_TA* Scorer);
void UpdateActiveMusicStinger(class APRI_TA* Player);
void UpdateReverbEnvironments();

```

```

void UpdateSoundScoreStatus();
void UpdateAnthemSoundState(unsigned long blsPlaying);
void UpdateSoundGameMode();
void UpdateSoundState();
void HandleMusicPlayerStateChanged(class UGameplayMusicPlayer_TA* InGameMusicPlayer,
uint8_t NewState);
void InitBackgroundMusic();
void HandleCountdownTimeUpdated(class AGameEvent_TA* InGameEvent, int32_t Seconds);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void MusicEventEnd();
bool IsMusicEventActive();
void StopEventMusic();
void MusicEventStart();
void StopInGameMusic();
void StartInGameMusic();
void HandleMatchEnded(class AGameEvent_Soccar_TA* InGameEvent);
void ResetAnthemAudio();
void StopAllStingers();
void StopGoalScoredStinger();
void PlayGoalScoredStinger();
void PlayGoalScoredSounds();
void PlayMutableMusicStinger(class UAkSoundCue* Stinger);
void PlayMusicStinger(class UAkSoundCue* Stinger);
bool IsSoundtrackOn();
void HandleMusicStingersSettingChanged();
bool IsStreamerSafe(class UProductAsset_MusicStingers_TA* MusicStinger);
bool IsMusicStingersActive();
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleStatEvent(class AGameEvent_Soccar_TA* InGameEvent, class UStatEvent_TA*
StatEvent);
void HandleGameStateChanged(class AGameEvent_TA* InGameEvent);
void PlaySound(class UAkSoundCue* Sound);
void HandlePylonSet();
void HandleConfigChanged();
void RegisterGameEventCallbacks(class AGameEvent_Soccar_TA* InSoccarGame);
void StreamerSafeConfigChanged();
void eventDetached();
void eventAttached();
static bool ShouldAttach();
void EventAnthemSoundStateUpdated(unsigned long bSoundState);
void EventActiveStingerUpdated(class UProductAsset_MusicStingers_TA* Stinger);
void EventReflectionsManagerSet(class UArenaReflectionsManager_TA*
NewReflectionsManager);
};

// Class TAGame._AutoTour_TA__OnReceivedBracket_0x2
// 0x0048 (0x0060 - 0x00A8)
class U_AutoTour_TA__OnReceivedBracket_0x2 : public UObject
{
public:
    struct FUniqueNetId           PlayerID;                      // 0x0060 (0x0048)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__AutoTour_TA__OnReceivedBracket_0x2");
}

return uClassPointer;
};

bool
__AutoTour_TA__OnReceivedBracket_0x2__AutoTour_TA__OnReceivedBracket_0x1_0x1(struct FTourPlayer P);
bool __AutoTour_TA__OnReceivedBracket_0x1(struct FTourTeam T);
bool __AutoTour_TA__OnReceivedBracket_0x2(struct FTourPlayer P);
};

// Class TAGame._TourTypes_TA
// 0x0020 (0x0060 - 0x0080)
class U_TourTypes_TA : public UObject
{
public:
TArray<struct FName> DisabledMutatorCategories; // 0x0060
(0x0010) [0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> DisabledMutatorSettings; // 0x0070
(0x0010) [0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._TourTypes_TA");
}

return uClassPointer;
};

static TArray<struct FName> __TourTypes_TA__StripDisabledGameTags_0x3(class UGameSettingCategory_X* GSC);
static struct FName __TourTypes_TA__StripDisabledGameTags_0x4(class UGameSetting_X* GS);
static bool __TourTypes_TA__StripDisabledGameTags_0x2(class UGameSettingCategory_X* GSC);
static struct FName __TourTypes_TA__StripDisabledGameTags_0x1(class FString SettingName);
static class FString StripDisabledGameTags(class FString GameTags, TArray<class UGameSettingCategory_X*>& GameSettingCategories);
};

```

```

// Class TAGame.AutoTour_TA
// 0x0128 (0x0060 - 0x0188)
class UAutoTour_TA : public UObject
{
public:
TArray<struct FScheduledTournament> Schedules; // 0x0060
(0x0010) [0x0009004000400000] (CPF_NeedCtorLink)
TArray<struct FTournamentWeek> CycleResults; // 0x0070
(0x0010) [0x0009004000402000] (CPF_Transient | CPF_NeedCtorLink)
uint64_t IneligibleScheduleID; // 0x0080 (0x0008)
[0x0009004000000000]
class FString RegionID; // 0x0088 (0x0010)
[0x000900000400000] (CPF_NeedCtorLink)
uint64_t CycleID; // 0x0098 (0x0008)
[0x001000000000000]
uint64_t CycleEndTime; // 0x00A0 (0x0008)
[0x001000040000000] (CPF>EditInlineNotify)
int32_t WeekID; // 0x00A8 (0x0004)
[0x001000040000000] (CPF>EditInlineNotify)
uint64_t WeekEndTime; // 0x00B0 (0x0008)
[0x001000040000000] (CPF>EditInlineNotify)
TArray<struct FShopCurrencyInfo> WeeklyCurrencies; // 0x00B8
(0x0010) [0x001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<int32_t> MaxTierScores; // 0x00C8 (0x0010)
[0x001004000402000] (CPF_Transient | CPF_NeedCtorLink)
float DataRefreshTime; // 0x00D8 (0x0004)
[0x001000000000000]
float LastSyncTime; // 0x00DC (0x0004)
[0x001000000000000]
class UAutoTourConfig_TA* Config; // 0x00E0 (0x0008)
[0x001800000000000]
class UOnlineGame_X* OnlineGame; // 0x00E8 (0x0008)
[0x001800000000000]
class UOnlineGameTournaments_TA* Tournaments; // 0x00F0
(0x0008) [0x00180000000001] (CPF>Edit)
class UTourList_TA* TourList; // 0x00F8 (0x0008)
[0x00100004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UAsyncTask* SyncCycleDataTask; // 0x0100 (0x0008)
[0x00100000002000] (CPF_Transient)
class UAsyncTask* SyncScheduleTask; // 0x0108 (0x0008)
[0x00100000002000] (CPF_Transient)
struct FScriptDelegate __EventTourCardsChanged__Delegate; // 0x0110
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __Schedules__ChangeNotify; // 0x0128
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __CycleResults__ChangeNotify; // 0x0140
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __IneligibleScheduleID__ChangeNotify; // 0x0158
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __RegionID__ChangeNotify; // 0x0170
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AutoTour_TA");
}

return uClassPointer;
};

void __AutoTour_TA__Construct_0x1(class UTourSubscriptions_TA* S, class UError* E);
void __AutoTour_TA__SetScheduleRegionID_0x1(class URPC_TourSetScheduleRegion_TA* RPC);
void __AutoTour_TA__SyncSchedule_0x1(class URPC_AutoTour_GetSchedule_TA* RPC);
TArray<class UTourSettings_TA*> __AutoTour_TA__SetSchedules_0x1(struct
FScheduledTournament S);
bool __AutoTour_TA__UpdateIneligibleTournament_0x1(class UTourSettings_TA* T);
void __RegionID__ChangeNotifyFunc();
void __IneligibleScheduleID__ChangeNotifyFunc();
void __CycleResults__ChangeNotifyFunc();
void __Schedules__ChangeNotifyFunc();
void OnReceivedBracket(uint64_t InScheduleID, struct FTourBracket& Bracket);
void UpdateIneligibleTournament();
void HandleTourEventDeactivated(class UTourEvent_TA* Event, class UError* Error);
void HandleTourActivated(class UOnlineGameTournaments_TA* Tournament, class
UTourEvent_TA* Event);
void HandleTourFound(class UPsyNetService_AutoTourFound_TA* Service);
void SetWeeksResults(int32_t InWeekIndex, TArray<struct FTournamentResult>&
NewWeeksResults);
void HandleCycleSynced(class URPC_AutoTour_GetCycleData_TA* RPC);
void ClearScheduleRefreshTimer();
void SyncAllTournamentData();
bool RefreshTournamentData(unsigned long bForce);
void SetSchedules(TArray<struct FScheduledTournament>& InSchedules);
void SyncCycleData();
void SyncSchedule();
class UAsyncTask* SetScheduleRegionID(class FString InRegionID);
void ReceivedPingRegions(class UOnlineGameRegions_X* RegionsObj);
void HandleScheduleRegionIDReceived(class URPC_TourGetScheduleRegion_TA* RPC);
void SyncScheduleRegionID();
void HandlePsyNetLogin();
void eventConstruct();
void EventTourCardsChanged();
};

// Class TAGame.__AutoTour_TA__UpdateIneligibleTournament_0x2
// 0x0008 (0x0060 - 0x0068)
class U__AutoTour_TA__UpdateIneligibleTournament_0x2 : public UObject
{
public:
class UTourSettings_TA*           LastAutoTournament;          // 0x0060
(0x0008) [0x0001000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._AutoTour_TA_UpdateineligibleTournament_0x2");
}

return uClassPointer;
};

void __AutoTour_TA__UpdateineligibleTournament_0x2(struct FTourBracket B);
};

// Class TAGame.TourSettings_TA
// 0x0118 (0x0060 - 0x0178)
class UTourSettings_TA : public UObject
{
public:
uint64_t Id; // 0x0060 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t ScheduleID; // 0x0068 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
class FString Title; // 0x0070 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString CreatorName; // 0x0080 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
struct FUniqueNetId CreatorPlayerID; // 0x0090 (0x0048)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
uint64_t StartTime; // 0x00D8 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
uint64_t GenerateBracketTime; // 0x00E0 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t MaxBracketSize; // 0x00E8 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t TeamsRegistered; // 0x00EC (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t RankMin; // 0x00F0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t RankMax; // 0x00F4 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
class FString Region; // 0x00F8 (0x0010)
[0x0001000004000000] (CPF_NeedCtorLink)
class FString RegionName; // 0x0108 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString GameTags; // 0x0118 (0x0010)
[0x0001000004000000] (CPF_NeedCtorLink)
uint8_t GameMode; // 0x0128 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
uint8_t TieBreaker; // 0x0129 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
uint8_t SeedingType; // 0x012A (0x0001)

```

```

[0x0001000040000000] (CPF>EditInlineNotify)
int32_t TeamSize; // 0x012C (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
TArray<struct FName> DisabledMaps; // 0x0130 (0x0010)
[0x0001000000400000] (CPF>NeedCtorLink)
TArray<int32_t> SeriesRoundLengths; // 0x0140 (0x0010)
[0x0001000000400000] (CPF>NeedCtorLink)
int32_t SeriesLength; // 0x0150 (0x0004)
[0x0001000000000000]
int32_t FinalSeriesLength; // 0x0154 (0x0004)
[0x0001000000000000]
unsigned long bPublic : 1; // 0x0158 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
int32_t MinPlayersPerTeam; // 0x015C (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
uint64_t MatchExpireTimestamp; // 0x0160 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
TArray<class FString> Platforms; // 0x0168 (0x0010)
[0x0001000000400000] (CPF>NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourSettings_TA");
}

return uClassPointer;
};

void __TourSettings_TA__Init_0x2(class FString Original, class FString Sanitized);
void __TourSettings_TA__Init_0x1(class FString Original, class FString Sanitized);
bool __TourSettings_TA__HandleRegionsSynced_0x1(struct FGFxRegion R);
bool IsCrossPlatformPlayEnabled();
bool IsManualTournament();
void HandleRegionsSynced(class UGFxData_Regions_TA* RegionsData);
void UpdateRegion(class UGFxData_Regions_TA* RegionsData);
void SetTeamsRegistered(int32_t InValue);
void Init();
};

// Class TAGame.__Ball_TA__Explode_0x1
// 0x0010 (0x0060 - 0x0070)
class U_Ball_TA__Explode_0x1 : public UObject
{
public:
class UGoal_TA* ExplosionGoal; // 0x0060 (0x0008)
[0x0000000004080008] (CPF>ExportObject | CPF>Component | CPF>EditInline)
class APRI_TA* Scorer; // 0x0068 (0x0008)
[0x0000000000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Ball_TA__Explode_0x1");
}

return uClassPointer;
};

void __Ball_TA__Explode_0x1(class APRI_TA* _);
};

// Class TAGame.__BanSync_TA__GetBanMessage_0x1
// 0x0050 (0x0060 - 0x00B0)
class U__BanSync_TA__GetBanMessage_0x1 : public UObject
{
public:
uint8_t BanType; // 0x0060 (0x0001)
[0x0000000000000000]
struct FUniqueNetId PlayerID; // 0x0068 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__BanSync_TA__GetBanMessage_0x1");
}

return uClassPointer;
};

bool __BanSync_TA__GetBanMessage_0x1(class UBanMessage_X* M);
};

// Class TAGame.BanSync_TA
// 0x0028 (0x0060 - 0x0088)
class UBanSync_TA : public UObject
{
public:
TArray<class UBanMessage_X*> BanMessages; // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSyncedBanStatus__Delegate; // 0x0070
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BanSync_TA");
}

return uClassPointer;
};

bool __BanSync_TA__SyncAllPlayers_0x3(struct FUniqueNetId Id);
struct FUniqueNetId __BanSync_TA__SyncAllPlayers_0x2(class APRI_TA* P);
bool __BanSync_TA__SyncAllPlayers_0x1(class APRI_TA* P);
void __BanSync_TA__SetBanMessages_0x1(class UBanMessage_X* M);
static bool IsBanned(uint8_t BanType, struct FUniqueNetId PlayerID);
void Clear();
class UBanMessage_X* GetBanMessage(uint8_t BanType, struct FUniqueNetId PlayerID);
TArray<class UBanMessage_X*> GetPlayerBanMessages(struct FUniqueNetId PlayerID);
TArray<class UBanMessage_X*> GetBanMessageTypes(uint8_t BanType);
void SetBanMessages(TArray<class UBanMessage_X*>& Messages);
void HandleBanStatusResponse(class URPC_GetBanStatus_TA* RPC);
void SyncBanStatus(TArray<struct FUniqueNetId>& PlayerIds);
void SyncLocalPlayers();
void SyncLocalPlayersDelayed();
void HandleLocalPlayersChanged(class ULocalPlayer* NewPlayer);
void HandlePsyNetLogin();
void SyncAllPlayers();
void HandlePlayerAdded(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void eventConstruct();
void EventSyncedBanStatus(class UBanSync_TA* BanSync);
};

// Class TAGame.__BanSync_TA__GetBanMessageTypes_0x1
// 0x0001 (0x0060 - 0x0061)
class U__BanSync_TA__GetBanMessageTypes_0x1 : public UObject
{
public:
uint8_t BanType; // 0x0060 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__BanSync_TA__GetBanMessageTypes_0x1");
}

return uClassPointer;
};

```

```
bool __BanSync_TA__GetBanMessageTypes_0x1(class UBanMessage_X* M);
};

// Class TAGame.__BanSync_TA__GetPlayerBanMessages_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__BanSync_TA__GetPlayerBanMessages_0x1 : public UObject
{
public:
    struct FUniqueNetId           PlayerID;          // 0x0060 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__BanSync_TA__GetPlayerBanMessages_0x1");
        }

        return uClassPointer;
    }

    bool __BanSync_TA__GetPlayerBanMessages_0x1(class UBanMessage_X* M);
};

// Class TAGame.__BinaryUploader_TA__Upload_0x1
// 0x0008 (0x0060 - 0x0068)
class U__BinaryUploader_TA__Upload_0x1 : public UObject
{
public:
    class UWebRequest_X*          Request;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__BinaryUploader_TA__Upload_0x1");
        }

        return uClassPointer;
    }

    void __BinaryUploader_TA__Upload_0x1(class FString K, class FString V);
};

// Class TAGame.BinaryUploader_TA
```

```
// 0x0018 (0x0060 - 0x0078)
class UBinaryUploader_TA : public UObject
{
public:
TArray<uint8_t> Data; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UStringMap* Headers; // 0x0070 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BinaryUploader_TA");
}

return uClassPointer;
};

void Upload(class FString URL);
void SetHeader(class FString Key, class FString Value);
};

// Class TAGame.__BotConfig_TA__RefreshIDs_0x1
// 0x0004 (0x0060 - 0x0064)
class U__BotConfig_TA__RefreshIDs_0x1 : public UObject
{
public:
int32_t ProductID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__BotConfig_TA__RefreshIDs_0x1");
}

return uClassPointer;
};

bool __BotConfig_TA__RefreshIDs_0x2(int32_t Id);
bool __BotConfig_TA__RefreshIDs_0x1(int32_t Id);
};

// Class TAGame.BotConfig_TA
// 0x00C8 (0x0078 - 0x0140)
class UBotConfig_TA : public UOnlineConfig_X
```

```

{
public:
struct FInterpCurveFloat           BotSkillByPlayerSkillCurve;          // 0x0078 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FInterpCurveFloat           BotSkillCurveForFTEMatches;        // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
float                             BotBoostThreshold_vsAI;           // 0x00A8 (0x0004)
[0x0000000000000000]
float                             Skills[0x5];                      // 0x00AC (0x0014)
[0x0000000000000000]
TArray<int32_t>                  AvatarIDs;                     // 0x00C0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>                  BannerIDs;                     // 0x00D0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>                  UnselectedAvatarIDs;           // 0x00E0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>                  UnselectedBannerIDs;           // 0x00F0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>                  AllowedAvatarIDs;            // 0x0100 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t>                  AllowedBannerIDs;            // 0x0110 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t>                  IgnoredAvatarIDs;             // 0x0120 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t>                  IgnoredBannerIDs;            // 0x0130 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BotConfig_TA");
}

return uClassPointer;
};

static int32_t PickRandomUnusedProductID(TArray<int32_t>& AllIDs, TArray<int32_t>& UnselectedIDs);
int32_t PickUnusedBannerProductID();
int32_t PickUnusedAvatarProductID();
static TArray<int32_t> RefreshIDs(class UProductSlot_TA* Slot, TArray<int32_t>& AllowedIDs,
TArray<int32_t>& IgnoredIDs);
void Apply();
};

// Class TAGame.__CameraState_CarPreview_TA__GetSingleActorOrientation_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CameraState_CarPreview_TA__GetSingleActorOrientation_0x1 : public UObject
{
public:

```

```
class ACarPreviewActor_TA*           ViewTarget;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CameraState_CarPreview_TA__GetSingleActorOrientation_0x1");
}

return uClassPointer;
};

bool __CameraState_CarPreview_TA__GetSingleActorOrientation_0x1(struct FTargetCacheEntry
E);
};

// Class TAGame.CarPreviewActor_TA
// 0x01E0 (0x0268 - 0x0448)
class ACarPreviewActor_TA : public AActor
{
public:
int32_t           ControllerId;          // 0x0268 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UProductLoader_TA*           ProductLoader;        // 0x0270 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCarMeshComponent_TA*         CarMesh;            // 0x0278 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class ULightEnvironmentComponent*   LightEnvironment;    // 0x0280
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_EditInline)
TArray<class UCarPreviewAnim_TA*>   AttachmentPreviewAnims; // 
0x0288 (0x0010) [0x000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_NeedCtorLink | CPF_EditInline)
TArray<class UCarPreviewAnim_TA*>   WheelAnims;          // 0x0298
(0x0010) [0x000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_NeedCtorLink | CPF_EditInline)
class UCarPreviewAnim_TA*           BodyAnim;            // 0x02A8 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCarPreviewAnim_TA*           EngineAudioAnim;     // 0x02B0
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_EditInline)
class UCarPreviewAnim_TA*           EngineAudioAnimInstance; // 0x02B8
(0x0008) [0x000000400408008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
float                            SupersonicFakeSpeed;   // 0x02C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class ATurnTableActor_TA*           TurntableActor;      // 0x02C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FLoadoutData               Loadout;             // 0x02D0 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
```

```

struct FLoadoutData           OldLoadout;          // 0x0310 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UProductAsset_TA*>   PreviewProducts;    // 0x0350
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class AFXActor_Boost_TA*       BoostFX;           // 0x0360 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UProductSlot_TA*        PreviewSlot;         // 0x0368 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ATeam_TA*               TeamArchetype;      // 0x0370 (0x0008)
[0x0000004000002000] (CPF_Transient)
class FString                 PlayerName;          // 0x0378 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
class AFXActor_X*              BodyFX;             // 0x0388 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UProductAsset_Boost_TA*  BoostAsset;          // 0x0390 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UProductAsset_SupersonicTrail_TA*  SupersonicAsset; // 0x0398
(0x0008) [0x0000004000002000] (CPF_Transient)
unsigned long                  bLockLoadout : 1;     // 0x03A0 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long                  bSpinWheelsFromBoost : 1; // 0x03A0 (0x0004)
[0x0000004000002000] [0x00000002] (CPF_Transient)
unsigned long                  bAddToCarPreviewList : 1; // 0x03A0 (0x0004)
[0x0000000000000000] [0x00000004]
TArray<class AFXActor_TA*>      SupersonicFX;        // 0x03A8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UProductAsset_EngineAudio_TA* EngineAudioAsset; // 0x03B8
(0x0008) [0x0000004000002000] (CPF_Transient)
class UEngineAudioPreviewBase_TA* EngineAudioPreview; // 0x03C0
(0x0008) [0x0000004000002000] (CPF_Transient)
class UProductAttribute_PreviewWheelSpin_TA* WheelSpinAttribute; // 0x03C8 (0x0008) [0x0000004000002000] (CPF_Transient)
int32_t                        PreviewTeam;         // 0x03D0 (0x0004)
[0x0000000000000000] (CPF_Transient)
int32_t                        LinkedControllerId; // 0x03D4 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X*              RootFXActor;        // 0x03D8 (0x0008)
[0x0000000000000000]
class UAkParamGroup*           Ak;                // 0x03E0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
struct FScriptDelegate          __EventInitializeComponents__Delegate; // 0x03E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __ProductApplicatorDelegate__Delegate; // 0x0400
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __EventCarSetupFinished__Delegate; // 0x0418
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __EventAnimTreeInitialized__Delegate; // 0x0430
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarPreviewActor_TA");
}

return uClassPointer;
};

void __CarPreviewActor_TA__PostBeginPlay_0x1(class UCarMeshComponentBase_TA* _);
bool __CarPreviewActor_TA__ActivatePreviewSlot_0x1(class UProductAsset_TA* X);
void RefreshLoadout(class UProfile_TA* Profile);
int32_t GetTeamIndex(class UProfile_TA* Profile);
void UpdateTeamFXState();
void SetTeamIndex(class UProfile_TA* Profile, int32_t TeamIndex);
class ULoadoutSet_TA* GetLoadoutSet(class UProfile_TA* Profile);
void SetLoadoutSet(class ULoadoutSet_TA* InLoadoutSet, int32_t InTeamIndex);
void EnableTick(unsigned long bEnable);
void DestroyTurnTableActor();
void ResetColors();
void SetLockLoadout(unsigned long bInLock);
void BuildOnlineLoadout(struct FLoadoutData& InLoadout, struct FClientLoadoutOnlineData& OnlineLoadout, TArray<int32_t>& OutOnlineLoadout);
void UpdateParticlesFakeVelocity();
struct FRotator GetSMRotation();
void SetTurnTableInput(float YawInput, unsigned long bGamepad);
void ApplyTurtableBase();
void SetTurnTableActor(class ATurnTableActor_TA* InTurnTableActor, struct FRotator StartRotation);
void FindTurnTableActor();
void eventOnOwnerChanged();
void Tick(float DeltaTime);
void SetPreviewHidden(unsigned long bHide);
void SetPlayerName(class FString InPlayerName);
void AnimateBody();
void AnimateWheels();
void SetStartAndEndWheelAnimIndices(int32_t& OutStartIdx, int32_t& OutEndIdx);
void AnimateAttachment(class UProductSlot_TA* Slot, class UPrimitiveComponent* AttachmentComponent, struct FName CustomAnimName);
void AnimateAntenna(class UAntennaComponent_TA* AntennaArchetype);
class UPrimitiveComponent* FindAttachmentComponent(class UProductSlot_TA* Slot, struct FProductAttachment& AttachStruct);
void AnimateSlotChange(class UProductAsset_TA* Product);
class UActorComponent* FindAttachmentByArchetype(class UActorComponent* InArchetype);
class UStaticMeshComponent* FindSMAAttachmentByAsset(class UStaticMesh* Mesh);
class USkeletalMeshComponent* FindSKAttachmentByAsset(class USkeletalMesh* Mesh);
void UpdateTranslations();
void ClearBodyFX();
void ClearBoostFX();
void ClearComponents();
void InitSupersonicFX();
void InitEngineAudioFX();
void InitBoostFX();
void InitBodyFX();
void InitRootFXActor();
```

```

void InitComponents();
void HandleAllProductsLoaded(class UProductLoader_TA* Loader);
void SetBoostGlow(float Glow);
void UpdateBoostGlow();
void DeactivatePreviewSlot();
void ActivatePreviewSlot();
void SetPreviewSlot(class UProductSlot_TA* InSlot);
bool ChangeCarPart(int32_t SlotIndex, int32_t ProductID, struct FProductInstanceId OnlineID,
int32_t TeamPaintIndex);
static void ForceGarbageCollection();
void ForceSetLoadout(unsigned long bBuildOnlineLoadout, struct FLoadoutData& InLoadout);
void SetLoadout(struct FLoadoutData& InLoadout);
void eventPostInitAnimTree(class USkeletalMeshComponent* SkelComp);
void eventDestroyed();
void eventPostBeginPlay();
void EventAnimTreeInitialized();
void EventCarSetupFinished(class ACarPreviewActor_TA* PreviewActor);
void ProductApplicatorDelegate(class UProductAsset_TA* Asset);
void EventInitializeComponents(class ACarPreviewActor_TA* PreviewActor);
};

// Class TAGame.CameraState_CarPreview_TA
// 0x0084 (0x008C - 0x0110)
class UCameraState_CarPreview_TA : public UCameraState_X
{
public:
TArray<float> SideOffsets; // 0x0090 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FCameraOrientation> ViewOffsets; // 0x00A0
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float BlendRate; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinPitch; // 0x00B4 (0x0004)
[0x0000000000000000]
float MaxPitch; // 0x00B8 (0x0004)
[0x0000000000000000]
struct FCameraOrientation CurrentOrientation; // 0x00BC (0x002C)
[0x0000000000002000] (CPF_Transient)
float CurrentSideOffset; // 0x00E8 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FRotator RotationOffset; // 0x00EC (0x000C)
[0x0000000000002000] (CPF_Transient)
class UCameraState_PremiumGarage_TA* PrevPremiumGarageState; //
0x00F8 (0x0008) [0x0000000000002000] (CPF_Transient)
TArray<struct FTargetCacheEntry> TargetsCache; // 0x0100
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.CameraState_CarPreview_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void AdjustOutPOV CameraRotation(struct FRotator AddedRotator, struct FCameraOrientation& OutPOV);
void OnSnap();
void PreviewFadeIn();
void PreviewFadeOut(class UCameraState_CarPreview_TA* InPrevState);
void BeginCameraState();
void GetAccumulatedOrientation(struct FVector& out_Location, struct FRotator& out_Rotation,
int32_t& NumPreviewActors);
void GetActorOrientation(struct FCameraOrientation& OutPOV, float& OutSideOffset);
void GetSingleActorOrientation(class ACarPreviewActor_TA* ViewTarget, struct FVector& OutLoc, struct FRotator& OutRot);
void UpdateCachedTarget(class ACarPreviewActor_TA* Target, struct FVector Location, struct FRotator Rotation);
bool IsTargetCached(class ACarPreviewActor_TA* Target);
};

// Class TAGame.CameraState_PremiumGarage_TA
// 0x0078 (0x0110 - 0x0188)
class UCameraState_PremiumGarage_TA : public UCameraState_CarPreview_TA
{
public:
struct FViewTargetTransitionParams           BetweenPremiumCameraBlend;          //
0x0110 (0x0010) [0x0000000000000000]
float                                         FadeInFromBlackTime;             // 0x0120 (0x0004)
[0x0000000000000000]
float                                         FadeToBlackTime;                // 0x0124 (0x0004)
[0x0000000000000000]
float                                         PitchDeadzone;                 // 0x0128 (0x0004)
[0x0000000000000000]
float                                         PitchSensitivity;              // 0x012C (0x0004)
[0x0000000000000000]
struct FInterpCurveFloat                   PitchAutocorrectCurve;          // 0x0130 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
float                                         PitchMaxSpeed;                 // 0x0148 (0x0004)
[0x0000000000000000]
float                                         PitchBlendbackModifier;        // 0x014C (0x0004)
[0x0000000000000000]
float                                         PitchBlendbackMin;              // 0x0150 (0x0004)
[0x0000000000000000]
struct FInterpCurveFloat                   CameraReturnBlendCurve;         // 0x0158
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
float                                         LastAdjustTime;                // 0x0170 (0x0004)
[0x0000000000002000] (CPF_Transient)
class ATurnTableActor_TA*                  TurntableActor;                 // 0x0178 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGFxShell_TA*                      Shell;                         // 0x0180 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_PremiumGarage_TA");
}

return uClassPointer;
};

void UpdateAllViewTargetRotation(float DeltaTime);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void AdjustOutPOV CameraRotation(struct FRotator AddedRotator, struct FCameraOrientation& OutPOV);
float GetCameraCurveSpeed();
void ManuallyAdjustCar(unsigned long bEnable);
void BeginCameraState();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
void GetActorOrientation(struct FCameraOrientation& OutPOV, float& OutSideOffset);
void GetAccumulatedOrientation(struct FVector& out_Location, struct FRotator& out_Rotation,
int32_t& NumPreviewActors);
};

// Class TAGame.__CameraState_CarPreview_TA__IsTargetCached_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CameraState_CarPreview_TA__IsTargetCached_0x1 : public UObject
{
public:
class ACarPreviewActor_TA* Target; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CameraState_CarPreview_TA__IsTargetCached_0x1");
}

return uClassPointer;
};

bool __CameraState_CarPreview_TA__IsTargetCached_0x1(struct FTargetCacheEntry E);
};

// Class TAGame.__CameraState_PodiumSpotlight_Knockout_TA__GetFocusCenter_0x1
// 0x000C (0x0060 - 0x006C)
```

```

class U__CameraState_PodiumSpotlight_Knockout_TA__GetFocusCenter_0x1 : public UObject
{
public:
    struct FVector FocusCenter; // 0x0060 (0x000C)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__CameraState_PodiumSpotlight_Knockout_TA__GetFocusCenter_0x1");
        }

        return uClassPointer;
    };

void __CameraState_PodiumSpotlight_Knockout_TA__GetFocusCenter_0x1(struct
FCachedPodiumCar C);
};

// Class TAGame.CameraState_PodiumSpotlight_TA
// 0x0044 (0x008C - 0x00D0)
class UCameraState_PodiumSpotlight_TA : public UCameraState_X
{
public:
    float CameraDistance; // 0x0090 (0x0004)
    [0x0000000000000000] (CPF_Transient)
    float CameraOffsetYawDeg; // 0x0094 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float CameraOffsetPitchDeg; // 0x0098 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float XOffsetDistanceStart; // 0x009C (0x0004)
    [0x0000000000000002] (CPF_Const)
    float XOffsetDistanceEnd; // 0x00A0 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float XOffsetDollySpeed; // 0x00A4 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float DefaultFOV; // 0x00A8 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float FadeInDuration; // 0x00AC (0x0004)
    [0x0000000000000002] (CPF_Const)
    float CameraZOffset; // 0x00B0 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float AnimDistanceOffset; // 0x00B4 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float AnimPitchOffset; // 0x00B8 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float AnimZOffset; // 0x00BC (0x0004)
    [0x0000000000000002] (CPF_Const)
    TArray<struct FCachedPodiumCar> CachedCars; // 0x00C0
}

```

(0x0010) [0x0000000000402000] (CPF\_Transient | CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.CameraState_PodiumSpotlight_TA");  
}  
  
return uClassPointer;  
};  
  
bool __CameraState_PodiumSpotlight_TA__UpdatePOV_0x1(struct FCachedPodiumCar CC);  
void UpdateCachedCarInfo(class AGameEvent_TA* GameEvent);  
struct FVector GetFocusCenter(struct FVector FieldCenter);  
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);  
void BeginCameraState();  
bool ShouldExecute();  
};  
  
// Class TAGame.CameraState_PodiumSpotlight_Knockout_TA  
// 0x0000 (0x00D0 - 0x00D0)  
class UCameraState_PodiumSpotlight_Knockout_TA : public UCameraState_PodiumSpotlight_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.CameraState_PodiumSpotlight_Knockout_TA");  
}  
  
return uClassPointer;  
};  
  
struct FVector GetFocusCenter(struct FVector FieldCenter);  
};  
  
// Class TAGame.__CameraState_PodiumSpotlight_TA__GetFocusCenter_0x1  
// 0x0018 (0x0060 - 0x0078)  
class U__CameraState_PodiumSpotlight_TA__GetFocusCenter_0x1 : public UObject  
{  
public:  
struct FVector FocusCenter; // 0x0060 (0x000C)  
[0x0000000000000000]  
struct FVector FieldCenter; // 0x006C (0x000C)
```

```

[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._CameraState_PodiumSpotlight_TA__GetFocusCenter_0x1");
}

return uClassPointer;
};

void __CameraState_PodiumSpotlight_TA__GetFocusCenter_0x1(struct FCachedPodiumCar C);
};

// Class TAGame._Car_KnockOut_TA__AllowCarComponentActivate_0x1
// 0x0008 (0x0060 - 0x0068)
class U_Car_KnockOut_TA__AllowCarComponentActivate_0x1 : public UObject
{
public:
class ACarComponent_TA*           CarComponent;           // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._Car_KnockOut_TA__AllowCarComponentActivate_0x1");
}

return uClassPointer;
};

bool __Car_KnockOut_TA__AllowCarComponentActivate_0x1(class ACarComponent_TA* C);
};

// Class TAGame.Car_KnockOut_TA
// 0x0238 (0x0B4860 - 0xD980)
class ACar_KnockOut_TA : public ACar_TA
{
public:
class APRI_KnockOut_TA*          PRI_KO;           // 0x0B4860 (0x0008)
[0x0001004000002000] (CPF_Transient)
struct FName                      ReplicatedStateName; // 0x0B5068 (0x0008)
[0x0001000100002020] (CPF_Net | CPF_Transient)
uint8_t                           ReplicatedStateChanged; // 0x0B5870 (0x0001)

```

```
[0x0001000100002020] (CPF_Net | CPF_Transient)
struct FlmpulseData           ReplicatedImpulse;          // 0x0B5C74 (0x0008)
[0x0001000100002020] (CPF_Net | CPF_Transient)
struct FVector                HitImpulse;             // 0x0B647C (0x000C)
[0x0001000000002000] (CPF_Transient)
float                         RespawnImmunityTime;    // 0x0B7088 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         StunTimeHitScale;      // 0x0B748C (0x0004)
[0x0001000000000002] (CPF_Const)
float                         AttackRadius;          // 0x0B7890 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         AttackYExtent;        // 0x0B7C94 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         AttackZExtent;        // 0x0B980 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         MaxAttackDegrees;     // 0x0B849C (0x0004)
[0x0001000000000002] (CPF_Const)
float                         DefaultAttackDamage; // 0x0B88A0 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         AttackLingerTime;     // 0x0B8CA4 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         GrabbingLingerTime; // 0x0B90A8 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         MinGrabTimeBeforeThrow; // 0x0B94AC (0x0004)
[0x0001000000000002] (CPF_Const)
float                         BlockTime;            // 0x0B98B0 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         GrabThrowTime;       // 0x0B9CB4 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         GrabStunTime;        // 0x0BA0B8 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         DodgeThrowImpulse;   // 0x0BA4BC (0x0004)
[0x0001000000000002] (CPF_Const)
float                         DoubleJumpThrowImpulse; // 0x0BA8C0 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         LightGrabTime;       // 0x0BAC4 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         HeavyGrabTime;       // 0x0BB0C8 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         AttackHitKnockbackScale; // 0x0B84CC (0x0004)
[0x0001000000000002] (CPF_Const)
float                         MinHitImpulseSpeed;  // 0x0B8D0 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         LightHitImpulseScale; // 0x0BCD4 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         HeavyHitImpulseScale; // 0x0BC0D8 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         SuddenKOImpulse;     // 0x0BDC4 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         SuddenKOTThrowTime; // 0x0BC8E0 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         LightHitPitchDegrees; // 0x0Bcce4 (0x0004)
[0x0001000000000002] (CPF_Const)
float                         HeavyHitPitchDegrees; // 0x0BD0E8 (0x0004)
```

```

[0x0001000000000002] (CPF_Const)
float SuddenKOHitPitchDegrees; // 0x0BD4EC (0x0004)
[0x0001000000000002] (CPF_Const)
float AttackBeginDelay; // 0x0BD8F0 (0x0004)
[0x0001000000000002] (CPF_Const)
float HitTime; // 0x0BDCF4 (0x0004)
[0x0001000000000002] (CPF_Const)
float AttackStunTorqueScale; // 0x0BE0F8 (0x0004)
[0x0001000000000002] (CPF_Const)
float TradeStunTorqueScale; // 0x0BE4FC (0x0004)
[0x0001000000000002] (CPF_Const)
float TradeStunTime; // 0x0BE8C00 (0x0004)
[0x0001000000000002] (CPF_Const)
float TradeStunImpulseScale; // 0x0BEC04 (0x0004)
[0x0001000000000002] (CPF_Const)
float TradeStunPitchDegrees; // 0x0BF0C08 (0x0004)
[0x0001000000000002] (CPF_Const)
float ThrowPitchOffset; // 0x0BF4C0C (0x0004)
[0x0001000000000002] (CPF_Const)
struct FVector GrabAttachOffset; // 0x0BF8C10 (0x000C)
[0x0001000000000002] (CPF_Const)
float ThrowBackwardVelocityScale; // 0x0C041C (0x0004)
[0x0001000000000002] (CPF_Const)
float TradeImpulseSpeed; // 0x0C208 (0x0004)
[0x0001000000000002] (CPF_Const)
float TorqueStunBeginDelay; // 0x0C0C24 (0x0004)
[0x0001000000000002] (CPF_Const)
float DamagePointsPerImpulseSpeed; // 0x0C1028 (0x0004)
[0x0001000000000002] (CPF_Const)
float ThrowVelocityTransferScale; // 0x0C142C (0x0004)
[0x0001000000000002] (CPF_Const)
class UTTarget_TA* AttachedTarget; // 0x0C1830 (0x0008)
[0x00010000408000A] (CPF_Const | CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<class ACarComponent_TA*> AttackComponents; // 0x0C2038
(0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
class ACarComponent_Torque_TA* TorqueComponent; // 0x0C3048
(0x0008) [0x000100000002000] (CPF_Transient)
TArray<struct FPendingHit> TakenHits; // 0x0C3850 (0x0010)
[0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
class ACar_KnockOut_TA* PendingGrabber; // 0x0C4860
(0x0008) [0x0001004000002000] (CPF_Transient)
class ACarComponent_TA* UsedAttackComponent; // 0x0C5068
(0x0008) [0x000100000002020] (CPF_Net | CPF_Transient)
float SuperSonicStartTime; // 0x0C5870 (0x0004)
[0x000100000002000] (CPF_Transient)
float SuperSonicTimeForHeavyAttack; // 0x0C5C74 (0x0004)
[0x0001000000000002] (CPF_Const)
float ReturnToSafezoneTime; // 0x0C6078 (0x0004)
[0x0001000000002000] (CPF_Transient)
float ThrowMashAlpha; // 0x0C647C (0x0004)
[0x0001000000002000] (CPF_Transient)
TArray<class APRI_TA*> AttackerPRIs; // 0x0C680 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
float ResetAttackPRIDelay; // 0x0C7890 (0x0004)

```

```

[0x0001000000000000]
class AFXActor_Knockout_Attack_TA*           FX_AttackArchetype;          // 0x0C980
(0x0008) [0x0001000000000002] (CPF_Const)
class AFXActor_Knockout_Attack_TA*           FX_GrabAttackArchetype;      //
0x0C88A0 (0x0008) [0x0001000000000002] (CPF_Const)
class AFXActor_TA*                          FX_HitAttackLightArchetype;   // 0x0C90A8
(0x0008) [0x0001000000000002] (CPF_Const)
class AFXActor_TA*                          FX_HitAttackHeavyArchetype; // 0x0C98B0
(0x0008) [0x0001000000000002] (CPF_Const)
class AFXActor_TA*                          FX_HitBlockArchetype;        // 0x0CA0B8
(0x0008) [0x0001000000000002] (CPF_Const)
class AFXActor_TA*                          FX_GrabHitArchetype;        // 0x0CA8C0
(0x0008) [0x0001000000000002] (CPF_Const)
struct FName                               DemoSoundAttachmentName;    // 0x0CB0C8
(0x0008) [0x0001000000000002] (CPF_Const)
class UAkSoundCue*                        DemoSoundOverride;          // 0x0CB8D0
(0x0008) [0x0001000000000002] (CPF_Const)
class AStunlock_TA*                      StunlockArchetype;          // 0x0CC0D8 (0x0008)
[0x0001000000000002] (CPF_Const)
class AStunlock_TA*                      Stunlock;                  // 0x0CC8E0 (0x0008)
[0x000100000002000] (CPF_Transient)
unsigned long                            bDrawDebug : 1;            // 0x0CD0E8 (0x0004)
[0x0001000000000002] [0x00000001] (CPF_Const)
class AFXActor_Knockout_Attack_TA*       FX_Attack;                 // 0x0CD8F0
(0x0008) [0x000100000002000] (CPF_Transient)
class ACar_KnockOut_TA*                  PendingCarToGrab;          // 0x0CE0F8
(0x0008) [0x000100000002000] (CPF_Transient)
class ACar_KnockOut_TA*                  CarPendingThrow;          // 0x0CE8D00
(0x0008) [0x000100000002000] (CPF_Transient)
struct FScriptDelegate                   __EventReplicatedStateChanged__Delegate; // 0x0CF0D08 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                   __EventDamageTaken__Delegate;   // 0x0D208
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                   __EventBlockTaken__Delegate; // 0x0D2038
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                   __EventGrabbedCarChanged__Delegate; // 0x0D3850 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                   __EventGrabbed__Delegate;     // 0x0D5068
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                   __EventThrown__Delegate;    // 0x0D680
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Car_KnockOut_TA");
}

return uClassPointer;
};

```

```
bool AttackComponentIsActive();
bool IsDodgingBackwards();
void BeginAttack();
struct FRotator GetAttackRotation();
void StopAttack();
class AFXActor_Knockout_Attack_TA* GetAttackFXArchetype();
void HandleAttackComponentChanged(class ACarComponent_TA* CarComponent);
void DetachCar();
void SubscribeToCarComponents();
void UnsubscribeFromCarComponents();
void HandleCarComponentChanged(class ACarComponent_TA* CarComponent);
void ApplyThrow();
void GrabbedCarChanged();
void StopGrabbing();
void OnHitComplete();
void BeginTorque();
void StopAttackStunned();
float GetStunTime();
float GetTorqueScale();
float GetMashTime();
void __Car_KnockOut_TA__DeactivateAttackComponents_0x2(class ACarComponent_TA* C);
uint8_t ApplyCarImpactForces(class ACar_TA* OtherCar, struct FVector HitLocation, struct FVector HitNormal);
bool ShouldDemolish(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal, uint8_t& Result);
bool CanOverrideExistingState();
bool CanBeGrabbed();
bool IsBeingGrabbed();
bool IsGrabStunned();
bool IsTradeStunned();
bool IsBlockStunned();
bool IsAttackStunned();
bool IsRespawning();
bool IsBlocking();
bool IsAttacking();
int32_t GetAttackDamage();
void UpdatePendingHits();
void HandleStunlockComplete();
void eventTick(float DeltaTime);
void SetDemoFXArchetypeOverrides(class AFXActor_X* OutDemoFXArchetype);
void ReturnToSafezoneTimeElapsed();
void SetInsideSafezone();
void SetOutsideSafezone(float SecondsToReturn);
bool IsOutsideSafezone();
void SetThrown(class ACar_KnockOut_TA* Thrower, struct FVector ThrowImpulse);
void StopGrabbed();
void StartGrabbed(float GrabTime);
void SetPendingGrab(class ACar_KnockOut_TA* Grabber);
bool CanViewTarget(class UTarget_TA* Target);
bool CanAttackTarget(class UTarget_TA* Target);
void FinishedPendingHits();
void ApplyPendingHits();
bool TookHitFrom(class ACar_KnockOut_TA* Car);
```

```

void AddTakenHit(struct FPendingHit NewHit);
bool ShouldDoHeavyAttack();
void eventOnSuperSonicChanged();
void GoToDefaultState();
void DeactivateComponents();
void DeactivateAttackComponents(TArray<class ACarComponent_TA*>& IgnoreComponents);
void eventOnGroundChanged();
void OnVehicleSetup();
void OnReplicatedHit();
void OnTakeHit(struct FName HitStateName, struct FVector ImpulseDirection, float
ImpulseSpeed, float PitchDegrees);
void OnTradedHit(TArray<class ACar_KnockOut_TA*>& Attackers);
void TakeDamage(struct FName HitState, int32_t DamageAmount, struct FVector
ImpulseDirection, float ImpulseSpeed, float PitchDegrees, unsigned long bHeavy, TArray<class
ACar_KnockOut_TA*>& Attackers);
void GotoReplicatedState(struct FName StateName);
void OnJumpPressed();
void ClearAttackerPRIs();
void SetAttackerPRI(class APRI_TA* Attacker);
void Demolish(class ARBActor_TA* Demolisher);
void FellOutOfWorld();
void OnPRIChanged();
void PostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventThrown(class ACar_KnockOut_TA* Car, class ACar_KnockOut_TA* Thrower);
void EventGrabbed(class ACar_KnockOut_TA* Car, class ACar_KnockOut_TA* Grabber);
void EventGrabbedCarChanged(class ACar_KnockOut_TA* Car, class ACar_KnockOut_TA*
GrabbedCar);
void EventBlockTaken(class ACar_KnockOut_TA* Car, TArray<class ACar_KnockOut_TA*>
Blockers, unsigned long bHeavy);
void EventDamageTaken(class ACar_KnockOut_TA* Car, TArray<class ACar_KnockOut_TA*>
Attackers, int32_t Amount, unsigned long bHeavy);
void EventReplicatedStateChanged(class ACar_KnockOut_TA* Car);
};


```

```

// Class TAGame.__Car_KnockOut_TA__DeactivateAttackComponents_0x1
// 0x0010 (0x0060 - 0x0070)
class U__Car_KnockOut_TA__DeactivateAttackComponents_0x1 : public UObject
{
public:
TArray<class ACarComponent_TA*>           IgnoreComponents;          // 0x0060
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Car_KnockOut_TA__DeactivateAttackComponents_0x1");
}
}
```

```
return uClassPointer;
};

bool __Car_KnockOut_TA__DeactivateAttackComponents_0x1(class ACarComponent_TA* C);
};

// Class TAGame.__Car_KnockOut_TA__TookHitFrom_0x1
// 0x0008 (0x0060 - 0x0068)
class U__Car_KnockOut_TA__TookHitFrom_0x1 : public UObject
{
public:
class ACar_KnockOut_TA*           Car;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Car_KnockOut_TA__TookHitFrom_0x1");
}

return uClassPointer;
};

bool __Car_KnockOut_TA__TookHitFrom_0x1(struct FPendingHit H);
};

// Class TAGame.__Car_KnockOut_TA__UpdatePendingHits_0x1
// 0x0008 (0x0060 - 0x0068)
class U__Car_KnockOut_TA__UpdatePendingHits_0x1 : public UObject
{
public:
class UTarget_TA*           Target;        // 0x0060 (0x0008)
[0x000100004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Car_KnockOut_TA__UpdatePendingHits_0x1");
}

return uClassPointer;
};

bool __Car_KnockOut_TA__UpdatePendingHits_0x1(class UTarget_TA* Attacked);
};
```

```

// Class TAGame.Target_TA
// 0x0020 (0x00A4 - 0x00C4)
class UTarget_TA : public UActorComponent_X
{
public:
    struct FVector LocalTranslation; // 0x00A8 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FVector WorldTranslation; // 0x00B4 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    int32_t Priority; // 0x00C0 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.Target_TA");
        }

        return uClassPointer;
    }

    void eventDetached();
    void eventAttached();
    struct FVector GetFocus();
};

// Class TAGame.__Car_TA__HandleTargetRemoved_0x1
// 0x0008 (0x0060 - 0x0068)
class U__Car_TA__HandleTargetRemoved_0x1 : public UObject
{
public:
    class UTarget_TA* Target; // 0x0060 (0x0008)
    [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__Car_TA__HandleTargetRemoved_0x1");
        }

        return uClassPointer;
    }

    bool __Car_TA__HandleTargetRemoved_0x1(class UTargetIndicator_TA* T);
};

```

```
// Class TAGame.__Car_TA__QueueDemolish_0x1
// 0x0030 (0x0060 - 0x0090)
class U_Car_TA__QueueDemolish_0x1 : public UObject
{
public:
struct FDemolishData           Data;          // 0x0060 (0x0028)
[0x0000000000000000]
class AFXActor_X*              GoalExplosionDemoFxArchetype; // 0x0088
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Car_TA__QueueDemolish_0x1");
}

return uClassPointer;
};

void __Car_TA__QueueDemolish_0x1(float _);

// Class TAGame.__Car_TA__QueueDemolish_Deprecated_0x1
// 0x0030 (0x0060 - 0x0090)
class U_Car_TA__QueueDemolish_Deprecated_0x1 : public UObject
{
public:
struct FDemolishData           Data;          // 0x0060 (0x0028)
[0x0000000000000000]
class AFXActor_X*              CustomDemoFX; // 0x0088 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Car_TA__QueueDemolish_Deprecated_0x1");
}

return uClassPointer;
};

void __Car_TA__QueueDemolish_Deprecated_0x1(float _);
};
```

```
// Class TAGame.__CarComponent_Boost_TA_SetUnlimitedBoostDelayed_0x1
// 0x0004 (0x0060 - 0x0064)
class U_CarComponent_Boost_TA_SetUnlimitedBoostDelayed_0x1 : public UObject
{
public:
unsigned long           Enabled : 1;           // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CarComponent_Boost_TA_SetUnlimitedBoostDelayed_0x1");
}

return uClassPointer;
};

void __CarComponent_Boost_TA_SetUnlimitedBoostDelayed_0x1();

// Class TAGame.__CarMeshComponentBase_TA_AddVisualAsset_0x2
// 0x0008 (0x0060 - 0x0068)
class U_CarMeshComponentBase_TA_AddVisualAsset_0x2 : public UObject
{
public:
class UProductAsset_TA*      Asset;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CarMeshComponentBase_TA_AddVisualAsset_0x2");
}

return uClassPointer;
};

void __CarMeshComponentBase_TA_AddVisualAsset_0x2(struct FProductAttachment A);

// Class TAGame.__CarMeshComponentBase_TA_ApplyPaintSettings_0x1
// 0x0038 (0x0060 - 0x0098)
class U_CarMeshComponentBase_TA_ApplyPaintSettings_0x1 : public UObject
{
```

```
public:
struct FAssociativeMaterialParams           AdditionalParams;           // 0x0060
(0x0038) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._CarMeshComponentBase_TA__ApplyPaintSettings_0x1");
}

return uClassPointer;
};

bool __CarMeshComponentBase_TA__ApplyPaintSettings_0x1(class UMaterialInterface* X);
};

// Class TAGame._CarMeshComponentBase_TA__ApplyPaintToObject_0x1
// 0x0020 (0x0060 - 0x0080)
class U_CarMeshComponentBase_TA__ApplyPaintToObject_0x1 : public UObject
{
public:
class UProductPaint_TA*                  ProductPaint;             // 0x0060 (0x0008)
[0x0000000000000000]
class UProductAttribute_PaintSettings_TA* PaintSettings;          // 0x0068
(0x0008) [0x0000000000000000]
class UObject*                           ObjectToPaint;          // 0x0070 (0x0008)
[0x0000000000000000]
class UProductAsset_TA*                 Product;                // 0x0078 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._CarMeshComponentBase_TA__ApplyPaintToObject_0x1");
}

return uClassPointer;
};

void __CarMeshComponentBase_TA__ApplyPaintToObject_0x1();
};

// Class TAGame._CarMeshComponentBase_TA__AttachOrnament_0x1
// 0x0008 (0x0060 - 0x0068)
```

```

class U_CarMeshComponentBase_TA_AttachOrnament_0x1 : public UObject
{
public:
    class UAprilConfig_X*           AprilConfig;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame._CarMeshComponentBase_TA_AttachOrnament_0x1");
        }

        return uClassPointer;
    }

void __CarMeshComponentBase_TA_AttachOrnament_0x2(class UPrimitiveComponent*
PrimComp);
void __CarMeshComponentBase_TA_AttachOrnament_0x1(class UPrimitiveComponent*
PrimComp);
};

// Class TAGame._CarMeshComponentBase_TA_CreateWheelAttachment_0x1
// 0x0010 (0x0060 - 0x0070)
class U_CarMeshComponentBase_TA_CreateWheelAttachment_0x1 : public UObject
{
public:
    class UProductAsset_Wheel_TA*      Asset;              // 0x0060 (0x0008)
    [0x0000000000000000]
    class UPrimitiveComponent*         PrimComp;           // 0x0068 (0x0008)
    [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame._CarMeshComponentBase_TA_CreateWheelAttachment_0x1");
        }

        return uClassPointer;
    }

void __CarMeshComponentBase_TA_CreateWheelAttachment_0x1(class
UAttachmentBehavior_TA* Behavior);
};

```

```
// Class TAGame.__CarMeshComponentBase_TA__CreateWheelMesh_0x1
// 0x0010 (0x0060 - 0x0070)
class U__CarMeshComponentBase_TA__CreateWheelMesh_0x1 : public UObject
{
public:
class UProductAsset_Wheel_TA* Asset; // 0x0060 (0x0008)
[0x0000000000000000]
class UMeshComponent* WheelMesh; // 0x0068 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CarMeshComponentBase_TA__CreateWheelMesh_0x1");
}

return uClassPointer;
};

void __CarMeshComponentBase_TA__CreateWheelMesh_0x1(struct FWheelAttachment A);
};

// Class TAGame.__CarMeshComponentBase_TA__RemoveProductAttributeFromProduct_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CarMeshComponentBase_TA__RemoveProductAttributeFromProduct_0x1 : public
UObject
{
public:
class UClass* AttributeClassToRemove; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CarMeshComponentBase_TA__RemoveProductAttributeFromProduct_0x1");
}

return uClassPointer;
};

bool __CarMeshComponentBase_TA__RemoveProductAttributeFromProduct_0x1(class
UProductAttribute_TA* CurAttribute);
};
```

```

// Class TAGame.__CarPreviewActor_TA__UpdateTranslations_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CarPreviewActor_TA__UpdateTranslations_0x1 : public UObject
{
public:
class UProductAsset_Body_TA*           BodyAsset;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CarPreviewActor_TA__UpdateTranslations_0x1");
}

return uClassPointer;
};

void __CarPreviewActor_TA__UpdateTranslations_0x1(struct FWheelSkelControlSet WC);
};

// Class TAGame.CarPreviewActor_TA
// 0x01E0 (0x0268 - 0x0448)
class ACarPreviewActor_TA : public AActor
{
public:
int32_t           ControllerId;           // 0x0268 (0x0004)
[0x0000000000000001] (CPF_Edit)

class UProductLoader_TA*           ProductLoader;           // 0x0270 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCarMeshComponent_TA*         CarMesh;           // 0x0278 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class ULightEnvironmentComponent*   LightEnvironment;   // 0x0280
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

TArray<class UCarPreviewAnim_TA*>   AttachmentPreviewAnims;           // 0x0288 (0x0010) [0x000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
TArray<class UCarPreviewAnim_TA*>   WheelAnims;           // 0x0298
(0x0010) [0x000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
class UCarPreviewAnim_TA*           BodyAnim;           // 0x02A8 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCarPreviewAnim_TA*           EngineAudioAnim;   // 0x02B0
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCarPreviewAnim_TA*           EngineAudioAnimInstance; // 0x02B8
(0x0008) [0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
float                           SupersonicFakeSpeed; // 0x02C0 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

class ATurnTableActor_TA*           TurntableActor;           // 0x02C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FLoadoutData               Loadout;                // 0x02D0 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FLoadoutData               OldLoadout;             // 0x0310 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UProductAsset_TA*>   PreviewProducts;        // 0x0350
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class AFXActor_Boost_TA*          BoostFX;                // 0x0360 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UProductSlot_TA*           PreviewSlot;            // 0x0368 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ATeam_TA*                  TeamArchetype;         // 0x0370 (0x0008)
[0x0000004000002000] (CPF_Transient)
class FString                   PlayerName;              // 0x0378 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
class AFXActor_X*                BodyFX;                 // 0x0388 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UProductAsset_Boost_TA*    BoostAsset;              // 0x0390 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UProductAsset_SupersonicTrail_TA* SupersonicAsset; // 0x0398
(0x0008) [0x0000004000002000] (CPF_Transient)
unsigned long                    bLockLoadout : 1;      // 0x03A0 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long                    bSpinWheelsFromBoost : 1; // 0x03A0 (0x0004)
[0x0000004000002000] [0x00000002] (CPF_Transient)
unsigned long                    bAddToCarPreviewList : 1; // 0x03A0 (0x0004)
[0x0000000000000000] [0x00000004]
TArray<class AFXActor_TA*>       SupersonicFX;            // 0x03A8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UProductAsset_EngineAudio_TA* EngineAudioAsset; // 0x03B8
(0x0008) [0x0000004000002000] (CPF_Transient)
class UEngineAudioPreviewBase_TA* EngineAudioPreview; // 0x03C0
(0x0008) [0x0000004000002000] (CPF_Transient)
class UProductAttribute_PreviewWheelSpin_TA* WheelSpinAttribute; // 0x03C8 (0x0008) [0x0000004000002000] (CPF_Transient)
int32_t                          PreviewTeam;             // 0x03D0 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t                          LinkedControllerId; // 0x03D4 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X*                RootFXActor;             // 0x03D8 (0x0008)
[0x0000000000000000]
class UAkParamGroup*             Ak;                     // 0x03E0 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
struct FScriptDelegate           _EventInitializeComponents__Delegate; // 0x03E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           _ProductApplicatorDelegate__Delegate; // 0x0400
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           _EventCarSetupFinished__Delegate; // 0x0418
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           _EventAnimTreeInitialized__Delegate; // 0x0430
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

public:

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarPreviewActor_TA");
}

return uClassPointer;
};

void __CarPreviewActor_TA__PostBeginPlay_0x1(class UCarMeshComponentBase_TA* _);
bool __CarPreviewActor_TA__ActivatePreviewSlot_0x1(class UProductAsset_TA* X);
void RefreshLoadout(class UProfile_TA* Profile);
int32_t GetTeamIndex(class UProfile_TA* Profile);
void UpdateTeamFXState();
void SetTeamIndex(class UProfile_TA* Profile, int32_t TeamIndex);
class ULoadoutSet_TA* GetLoadoutSet(class UProfile_TA* Profile);
void SetLoadoutSet(class ULoadoutSet_TA* InLoadoutSet, int32_t InTeamIndex);
void EnableTick(unsigned long bEnable);
void DestroyTurnTableActor();
void ResetColors();
void SetLockLoadout(unsigned long bInLock);
void BuildOnlineLoadout(struct FLoadoutData& InLoadout, struct FClientLoadoutOnlineData& OnlineLoadout, TArray<int32_t>& OutOnlineLoadout);
void UpdateParticlesFakeVelocity();
struct FRotator GetSMRotation();
void SetTurnTableInput(float YawInput, unsigned long bGamepad);
void ApplyTurntableBase();
void SetTurnTableActor(class ATurnTableActor_TA* InTurnTableActor, struct FRotator StartRotation);
void FindTurnTableActor();
void eventOnOwnerChanged();
void Tick(float DeltaTime);
void SetPreviewHidden(unsigned long bHide);
void SetPlayerName(class FString InPlayerName);
void AnimateBody();
void AnimateWheels();
void SetStartAndEndWheelAnimIndices(int32_t& OutStartIdx, int32_t& OutEndIdx);
void AnimateAttachment(class UProductSlot_TA* Slot, class UPrimitiveComponent* AttachmentComponent, struct FName CustomAnimName);
void AnimateAntenna(class UAntennaComponent_TA* AntennaArchetype);
class UPrimitiveComponent* FindAttachmentComponent(class UProductSlot_TA* Slot, struct FProductAttachment& AttachStruct);
void AnimateSlotChange(class UProductAsset_TA* Product);
class UActorComponent* FindAttachmentByArchetype(class UActorComponent* InArchetype);
class UStaticMeshComponent* FindSMAAttachmentByAsset(class UStaticMesh* Mesh);
class USkeletalMeshComponent* FindSKAttachmentByAsset(class USkeletalMesh* Mesh);
void UpdateTranslations();
void ClearBodyFX();
void ClearBoostFX();
void ClearComponents();
void InitSupersonicFX();
```

```

void InitEngineAudioFX();
void InitBoostFX();
void InitBodyFX();
void InitRootFXActor();
void InitComponents();
void HandleAllProductsLoaded(class UProductLoader_TA* Loader);
void SetBoostGlow(float Glow);
void UpdateBoostGlow();
void DeactivatePreviewSlot();
void ActivatePreviewSlot();
void SetPreviewSlot(class UProductSlot_TA* InSlot);
bool ChangeCarPart(int32_t SlotIndex, int32_t ProductID, struct FProductInstanceID OnlineID,
int32_t TeamPaintIndex);
static void ForceGarbageCollection();
void ForceSetLoadout(unsigned long bBuildOnlineLoadout, struct FLoadoutData& InLoadout);
void SetLoadout(struct FLoadoutData& InLoadout);
void eventPostInitAnimTree(class USkeletalMeshComponent* SkelComp);
void eventDestroyed();
void eventPostBeginPlay();
void EventAnimTreeInitialized();
void EventCarSetupFinished(class ACarPreviewActor_TA* PreviewActor);
void ProductApplicatorDelegate(class UProductAsset_TA* Asset);
void EventInitializeComponents(class ACarPreviewActor_TA* PreviewActor);
};

// Class TAGame.__ChallengeDefault_TA__RequestImage_0x1
// 0x0010 (0x0060 - 0x0070)
class U__ChallengeDefault_TA__RequestImage_0x1 : public UObject
{
public:
class FString                               imageURL;                                // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ChallengeDefault_TA__RequestImage_0x1");
}

return uClassPointer;
};

void __ChallengeDefault_TA__RequestImage_0x1(class UTexture2DDynamic* Texture);
};

// Class TAGame.ListenObject_TA
// 0x0020 (0x0060 - 0x0080)
class UListenObject_TA : public UObject
{

```

```

public:
int32_t           Id;                      // 0x0060 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FScriptDelegate      __EventRemoved__Delegate;    // 0x0068
(0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ListenObject_TA");
}

return uClassPointer;
};

void Remove();
void HandleChallengeObjectRemoved(class UListenObject_TA* inObj);
void ListenForRemove(class UListenObject_TA* inObj);
void Init(class UListenObject_TA* inObj);
void EventRemoved(class UListenObject_TA* Obj);
};

// Class TAGame.ChallengeDefault_TA
// 0x0158 (0x0080 - 0x01D8)
class UChallengeDefault_TA : public UListenObject_TA
{
public:
class FString          Title;            // 0x0080 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString          Description;        // 0x0090 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t                GroupId;          // 0x00A0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                Points;           // 0x00A4 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
unsigned long          bIsAchievement : 1; // 0x00A8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long          bIsRepeatable : 1; // 0x00A8 (0x0004)
[0x0001000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long          bIsPremium : 1;   // 0x00A8 (0x0004)
[0x0001000040000000] [0x00000004] (CPF>EditInlineNotify)
unsigned long          bAutoClaimRewards : 1; // 0x00A8 (0x0004)
[0x0001000040000000] [0x00000008] (CPF>EditInlineNotify)
int32_t                XPUnlockLevel;     // 0x00AC (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                RepeatLimit;       // 0x00B0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
class FString          IconURL;          // 0x00B8 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
class FString          BackgroundURL;    // 0x00C8 (0x0010)

```

```

[0x0001000000400000] (CPF_NeedCtorLink)
class UChallengeReward_TA*          Rewards;           // 0x00D8 (0x0008)
[0x0001000000000000]
TArray<class UChallengeRequirement_TA*> Requirements; // 0x00E0
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class FString          NotifyAvailableTitle; // 0x00F0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          NotifyAvailableDescription; // 0x0100 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          NotifyAvailableImageURL; // 0x0110 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString          NotifyCompleteTitle; // 0x0120 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          NotifyCompleteDescription; // 0x0130 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          NotifyCompleteImageURL; // 0x0140 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class UTexture*        Icon;           // 0x0150 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
class UTexture*        Background; // 0x0158 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
class UTexture*        NotifyAvailableImage; // 0x0160 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
class UTexture*        NotifyCompleteImage; // 0x0168 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t               BackgroundColor; // 0x0170 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
struct FScriptDelegate _EventIconChanged__Delegate; // 0x0178
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventBackgroundChanged__Delegate; // 0x0190
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventNotifyAvailableImageChanged__Delegate; // 0x01A8 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventNotifyCompleteImageChanged__Delegate; // 0x01C0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeDefault_TA");
}

return uClassPointer;
};

void __ChallengeDefault_TA__Setup_0x1(class UChallengeRequirement_TA* R);
void HandleImageDownloaded(class FString URL, class UTexture2DDynamic* Texture);
void RequestImage(class FString ImageURL);
void RequestImages();
void Setup();

```

```

void EventNotifyCompleteImageChanged(class UChallengeDefault_TA* Challenge);
void EventNotifyAvailableImageChanged(class UChallengeDefault_TA* Challenge);
void EventBackgroundChanged(class UChallengeDefault_TA* Challenge);
void EventIconChanged(class UChallengeDefault_TA* Challenge);
};

// Class TAGame.__ChallengeManager_TA__BroadcastChallengeChanges_0x7
// 0x0020 (0x0060 - 0x0080)
class U__ChallengeManager_TA__BroadcastChallengeChanges_0x7 : public UObject
{
public:
class FString NotificationTitle; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString NotificationBody; // 0x0070 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ChallengeManager_TA__BroadcastChallengeChanges_0x7");
}

return uClassPointer;
};

void __ChallengeManager_TA__BroadcastChallengeChanges_0x7(class AGameEvent_TA* _);
};

// Class TAGame.ChallengeManager_TA
// 0x0110 (0x0060 - 0x0170)
class UChallengeManager_TA : public UObject
{
public:
TArray<class UChallenge_TA*> Challenges; // 0x0060 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<int32_t> ChallengeProgressSyncIDs; // 0x0070 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<int32_t> ClearNewInfoChallengeIDQueue; // 0x0080
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<class UChallenge_TA*> RemovedChallenges; // 0x0090
(0x0010) [0x0001004000400000] (CPF_NeedCtorLink)
TArray<int32_t> PendingNewChallengeIDs; // 0x00A0 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<int32_t> PendingRemovedChallengeIDs; // 0x00B0
(0x0010) [0x0001004000400000] (CPF_NeedCtorLink)
class FString NewChallengesTitle; // 0x00C0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString NewChallengesDesc; // 0x00D0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

```

```

class FString           ExpiredChallengesTitle;          // 0x00E0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           ExpiredChallengesDesc;         // 0x00F0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           MixedChallengesDesc;          // 0x0100 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate _EventChallengeAdded__Delegate;    // 0x0110
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventChallengeCompleted__Delegate; // 0x0128
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventChallengeRewardCollected__Delegate; // 0x0140
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventChallengeProgressChanged__Delegate; // 0x0158
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeManager_TA");
}

return uClassPointer;
};

void __ChallengeManager_TA__Construct_0x2(class UChallengeDefault_TA* C);
void __ChallengeManager_TA__Construct_0x1(class UChallengeDefaultManager_TA* _, class UChallengeDefault_TA* C);
void __ChallengeManager_TA__HandleChallengesReceived_0x1(struct FChallengeProgress P);
void __ChallengeManager_TA__HandleChallengeProgressPushed_0x1(struct FChallengeProgress P);
bool __ChallengeManager_TA__BroadcastChallengeChanges_0x6(class UChallenge_TA* C);
bool __ChallengeManager_TA__BroadcastChallengeChanges_0x5(class UChallenge_TA* C);
bool __ChallengeManager_TA__BroadcastChallengeChanges_0x4(class UChallenge_TA* C);
bool __ChallengeManager_TA__BroadcastChallengeChanges_0x3(class UChallenge_TA* C);
bool __ChallengeManager_TA__BroadcastChallengeChanges_0x2(class UChallenge_TA* C);
bool __ChallengeManager_TA__BroadcastChallengeChanges_0x1(class UChallenge_TA* C);
void __ChallengeManager_TA__RemoveChallenges_0x1(class UChallenge_TA* C);
void __ChallengeManager_TA__HandlePreLoadMap_0x1(class UChallenge_TA* C);
class ULocalPlayer_TA* GetLocalPlayer();
class APlayerControllerBase_TA* GetPC();
class UChallenge_TA* FindChallenge(int32_t Id);
void CollectChallengeReward(int32_t ChallengeID, TArray<class UOnlineProduct_TA*> OnlineProductRewards, struct FOnlineReward& Reward);
int32_t FindChallengeIndex(int32_t Id);
void HandlePreLoadMap(class FString MapName);
void HandleTrainingCompleted(class AGameEvent_Training_TA* Training);
void HandleTutorialCompleted(class UTutorial_TA* Tutorial);
void TriggerChallengeCompleteNotification(class UChallenge_TA* Challenge);
void NotifyChallengeCompleted(class UChallenge_TA* Challenge);
void OnChallengeComplete(class UChallenge_TA* Challenge);

```

```

void NotifyChallengeAvailable(class UChallenge_TA* Challenge);
void SetChallengeProgress(struct FChallengeProgress& Progress);
void RemoveChallenges();
void RemoveChallenge(int32_t Id);
class UChallenge_TA* AddChallenge(class UChallengeDefault_TA* ChallengeDefault);
void ClearPendingNewInfoChallenges();
void ClearNewInfoDelayed(int32_t ChallengeID);
void TriggerPopupNotification(class FString Title, class FString Body);
void BroadcastChallengeChanges();
void HandleChallengesPushed(TArray<int32_t> NewChallengeIDs, TArray<int32_t>
RemovedChallengeIDs);
void HandleChallengeProgressPushed(class UPsyNetService_ChallengeProgress_TA* Message);
void HandleChallengesReceived(class URPC_GetChallengeProgress_TA* RPC);
void SyncPendingChallengeProgress();
void eventConstruct();
void EventChallengeProgressChanged(class UChallengeManager_TA* Manager);
void EventChallengeRewardCollected(class UChallengeManager_TA* Manager, class
UChallenge_TA* Challenge, TArray<class UOnlineProduct_TA*> OnlineProductRewards, struct
FOnlineReward& Reward);
void EventChallengeCompleted(class UChallengeManager_TA* Manager, class UChallenge_TA*
Challenge);
void EventChallengeAdded(class UChallengeManager_TA* Manager, class UChallenge_TA*
Challenge);
};

// Class TAGame.__ChallengeManager_TA__FindChallengeIndex_0x1
// 0x0004 (0x0060 - 0x0064)
class U__ChallengeManager_TA__FindChallengeIndex_0x1 : public UObject
{
public:
int32_t                      Id;                           // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ChallengeManager_TA__FindChallengeIndex_0x1");
}

return uClassPointer;
};

bool __ChallengeManager_TA__FindChallengeIndex_0x1(class UChallenge_TA* C);
};

// Class TAGame.Challenge_TA
// 0x0020 (0x01D8 - 0x01F8)
class UChallenge_TA : public UChallengeDefault_TA
{

```

```

public:
unsigned long           bIsHidden : 1;           // 0x01D8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long           bRewardsAvailable : 1;    // 0x01D8 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long           bNotifyNewInfo : 1;      // 0x01D8 (0x0004)
[0x0001000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long           bComplete : 1;          // 0x01D8 (0x0004)
[0x0001000040000000] [0x00000008] (CPF_EditInlineNotify)
int32_t                CompleteCount;        // 0x01DC (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t                ProgressResetTimeUTC; // 0x01E0 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
TArray<class UChallengeRequirement_TA*> RequirementProgress; // 0x01E8 (0x0010) [0x0001000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Challenge_TA");
}

return uClassPointer;
};

void __Challenge_TA__OnTimeElapsed_0x1(class UChallengeRequirement_TA* Requirement);
void __Challenge_TA__ResetChangedProgress_0x1(class UChallengeRequirement_TA* R);
bool __Challenge_TA__ProgressHasChanged_0x1(class UChallengeRequirement_TA* P);
class UChallengeManager_TA* GetChallengeManager();
bool ProgressHasChanged();
void ResetChangedProgress();
void OnRewardsCollected(TArray<class UOnlineProduct_TA*> OnlineProductRewards, struct FOnlineReward& Reward);
void OnTimeElapsed();
bool IsAvailable();
void HandleChallengeObjectRemoved(class UListenObject_TA* ChallengeObject);
void HandleNotifyCompleteImageChanged(class UChallengeDefault_TA* Challenge);
void HandleNotifyAvailableImageChanged(class UChallengeDefault_TA* Challenge);
void HandleBackgroundChanged(class UChallengeDefault_TA* Challenge);
void HandleIconChanged(class UChallengeDefault_TA* Challenge);
void SetRequirementProgress(TArray<struct FChallengeRequirementProgress>& Progress);
void SetChallengeProgress(struct FChallengeProgress Progress);
void Init(class UListenObject_TA* ChallengeObject);
};

// Class TAGame.__ChallengeReward_TA__Init_0x1
// 0x0008 (0x0060 - 0x0068)
class U_ChallengeReward_TA_Init_0x1 : public UObject
{
public:

```

```
class UListenObject_TA*           ListenObj;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__ChallengeReward_TA__Init_0x1");
}

return uClassPointer;
};

void __ChallengeReward_TA__Init_0x1(class UChallengeCurrency_TA* C);
};

// Class TAGame.ChallengeCurrency_TA
// 0x0008 (0x0080 - 0x0088)
class UChallengeCurrency_TA : public UListenObject_TA
{
public:
int32_t             CurrencyID;          // 0x0080 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t             Amount;              // 0x0084 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeCurrency_TA");
}

return uClassPointer;
};

};

// Class TAGame.ChallengeReward_TA
// 0x0040 (0x0080 - 0x00C0)
class UChallengeReward_TA : public UListenObject_TA
{
public:
int32_t             XP;                 // 0x0080 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
TArray<class UChallengeCurrency_TA*>   Currency;          // 0x0088
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData>       Products;         // 0x0098 (0x0010)
```

```
[0x0001000000400000] (CPF_NeedCtorLink)
class FString ProductHashIDs; // 0x00A8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UOnlineProductStoreSet_TA* OnlineProductStoreSet; // 0x00B8
(0x0008) [0x000100004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeReward_TA");
}

return uClassPointer;
};

void Remove();
void Init(class UListenObject_TA* ListenObj);
};

// Class TAGame.__ChatFilterManager_TA__HandleLocalPlayerJoin_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ChatFilterManager_TA__HandleLocalPlayerJoin_0x1 : public UObject
{
public:
class ULocalPlayer_TA* JoiningPlayer; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ChatFilterManager_TA__HandleLocalPlayerJoin_0x1");
}

return uClassPointer;
};

void __ChatFilterManager_TA__HandleLocalPlayerJoin_0x1(class UOnlinePlayer_X* _);
};

// Class TAGame.ChatFilterManager_TA
// 0x0068 (0x0060 - 0x00C8)
class UChatFilterManager_TA : public UObject
{
public:
```

```

uint8_t MatchChatFilter; // 0x0060 (0x0001)
[0x0008008000000000]
uint8_t PartyChatFilter; // 0x0061 (0x0001)
[0x0008008000000000]
uint8_t VoiceChatFilter; // 0x0062 (0x0001)
[0x0008008000000000]

struct FScriptDelegate __GetLocalPlayers__Delegate; // 0x0068
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __MatchChatFilter__ChangeNotify; // 0x0080
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __PartyChatFilter__ChangeNotify; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __VoiceChatFilter__ChangeNotify; // 0x00B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChatFilterManager_TA");
}

return uClassPointer;
};

void __ChatFilterManager_TA__HandleSaveLoaded_0x4(class UEOSVoiceSettingsSave_TA*
UnloadedVoiceSettingsSave);
void __ChatFilterManager_TA__HandleSaveLoaded_0x3(class UEOSVoiceSettingsSave_TA*
LoadedVoiceSettingsSave);
void __ChatFilterManager_TA__HandleSaveLoaded_0x2(class UGameplaySettingsSave_TA*
UnloadedGameplaySettingsSave);
void __ChatFilterManager_TA__HandleSaveLoaded_0x1(class UGameplaySettingsSave_TA*
LoadedGameplaySettingsSave);
void __VoiceChatFilter__ChangeNotifyFunc();
void __PartyChatFilter__ChangeNotifyFunc();
void __MatchChatFilter__ChangeNotifyFunc();
void UpdatePermissions();
uint8_t GetMostRestrictive(uint8_t Filter1, uint8_t Filter2);
void HandleSaveLoaded(class USaveGameManager_TA* InSaveGameManager, class
USaveData_TA* InSaveData, class UError* InError);
void HandleLocalPlayerLeave(class ULocalPlayer_TA* LeavingPlayer);
void HandleLocalPlayerJoin(class ULocalPlayer_TA* JoiningPlayer);
void eventConstruct();
TArray<class ULocalPlayer_TA*> GetLocalPlayers();
};

// Class TAGame.__CinematicIntroSequence_TA__GetActionBinding_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CinematicIntroSequence_TA__GetActionBinding_0x1 : public UObject
{
public:

```

```

struct FName ActionName; // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._CinematicIntroSequence_TA__GetActionBinding_0x1");
}

return uClassPointer;
};

bool __CinematicIntroSequence_TA__GetActionBinding_0x2(struct FPlayerBinding B);
bool __CinematicIntroSequence_TA__GetActionBinding_0x1(struct FPlayerBinding B);
};

// Class TAGame.CinematicIntroSequence_TA
// 0x00C0 (0x0060 - 0x0120)
class UCinematicIntroSequence_TA : public UObject
{
public:
int32_t SequenceIndex; // 0x0060 (0x0004)
[0x0001000000000000]
class UGFxModal_X* MessageModal; // 0x0068 (0x0008)
[0x0001000000000000]
TArray<struct FName> ActionNames; // 0x0070 (0x0010)
[0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<struct FName> PressedKeys; // 0x0080 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FPressedAxis> PressedAxes; // 0x0090 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString LocalizationFile; // 0x00A0 (0x0010)
[0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString LocalizationGroup; // 0x00B0 (0x0010)
[0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString LocalizationKeyPrefix; // 0x00C0 (0x0010)
[0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
float AxisDeadzone; // 0x00D0 (0x0004)
[0x0001000000000002] (CPF_Const)
float AxisPressTime; // 0x00D4 (0x0004)
[0x0001000000000002] (CPF_Const)
float AutoFinishTime; // 0x00D8 (0x0004)
[0x0001000000000002] (CPF_Const)
TArray<struct FPlayerBinding> TrackedBindingsPC; // 0x00E0
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FPlayerBinding> TrackedBindingsGamepad; // 0x00F0
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class UInteraction* InputInteraction; // 0x0100 (0x0008)
[0x0001000000000000]

```

```
struct FScriptDelegate           __EventSequenceFinished__Delegate;      // 0x0108
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CinematicIntroSequence_TA");
}

return uClassPointer;
};

void __CinematicIntroSequence_TA__StartSequence_0x1(class UGFxShell_X* _);
bool __CinematicIntroSequence_TA__AllActionsPressed_0x1(struct FName Action);
void CloseMessageModal();
bool GetActionBinding(struct FName ActionName, struct FPlayerBinding& Out_Binding);
class FString GetActionKey(struct FName ActionName);
bool IsAxisPressed(struct FPlayerBinding Binding);
bool IsKeyPressed(struct FPlayerBinding Binding);
bool ActionIsPressed(struct FName ActionName);
bool AllActionsPressed();
void CheckAllKeysPressed();
bool NotifyKeyInput(int32_t ControllerId, struct FName Key, uint8_t EventType, float
AmountDepressed, unsigned long bGamepad);
bool NotifyAxisInput(int32_t ControllerId, struct FName Key, float delta, float DeltaTime, unsigned
long bGamepad);
class FString GetBodyText();
void RefreshMessage();
void SetTrackedBindings();
void EndSequence();
void StartSequence(int32_t InSequenceIndex);
void EventSequenceFinished(class UCinematicIntroSequence_TA* Seq);
};

// Class TAGame.__CinematicIntroSequence_TA_SetTrackedBindings_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CinematicIntroSequence_TA_SetTrackedBindings_0x1 : public UObject
{
public:
struct FName                  ActionName;                      // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
```

```

TAGame.__CinematicIntroSequence_TA__SetTrackedBindings_0x1");
}

return uClassPointer;
};

bool __CinematicIntroSequence_TA__SetTrackedBindings_0x2(struct FPlayerBinding B);
bool __CinematicIntroSequence_TA__SetTrackedBindings_0x1(struct FPlayerBinding B);
};

// Class TAGame.__ClubPersonas_TA__QueueRefresh_0x1
// 0x0008 (0x0060 - 0x0068)
class U_ClubPersonas_TA__QueueRefresh_0x1 : public UObject
{
public:
    class UPersona_TA*           Persona;          // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__ClubPersonas_TA__QueueRefresh_0x1");
        }
    }

    return uClassPointer;
};

void __ClubPersonas_TA__QueueRefresh_0x1();

// Class TAGame.ClubPersonas_TA
// 0x0028 (0x0060 - 0x0088)
class UClubPersonas_TA : public UObject
{
public:
    class UOnlineClubCache_X*      ClubCache;        // 0x0060 (0x0008)
    [0x0001800004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UOnlineGameWordFilter_X* WordFilter;       // 0x0068 (0x0008)
    [0x0001800000000000]
    TArray<class UPersonaClubData*> PersonaClubs;   // 0x0070
    (0x0010) [0x000100000400000] (CPF_NeedCtorLink)
    class UActionQueue_X*          ActionQueue;      // 0x0080 (0x0008)
    [0x0001000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
    }
}

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.ClubPersonas_TA");
}

return uClassPointer;
};

void ClubTagWithNameSanitized(class UPersonaClubData* PClub, class FString ClubTag, class
FString Sanitized);
void RefreshPersona(class UPersona_TA* Persona);
void QueueRefresh(class UPersona_TA* Persona);
void HandlePersonaAdded(class UPersonas_TA* Personas, class UPersona_TA* Persona);
void HandleClubUpdated(class UOnlineClubCache_X* Cache, class UClubDetails_X* ClubDetails);
void eventConstruct();
};

// Class TAGame.__ClubPersonas_TA__RefreshPersona_0x1
// 0x0018 (0x0060 - 0x0078)
class U__ClubPersonas_TA__RefreshPersona_0x1 : public UObject
{
public:
class UPersona_TA* Persona; // 0x0060 (0x0008)
[0x0001000000000000]
class UPersonaClubData* PClub; // 0x0068 (0x0008)
[0x0001000000000000]
class UClubDetails_X* Club; // 0x0070 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__ClubPersonas_TA__RefreshPersona_0x1");
}

return uClassPointer;
};

void __ClubPersonas_TA__RefreshPersona_0x2(class FString O, class FString S);
bool __ClubPersonas_TA__RefreshPersona_0x1(class UPersonaClubData* P);
};

// Class TAGame.PersonaClubData
// 0x0020 (0x0060 - 0x0080)
class UPersonaClubData : public UObject
{
public:
class UPersona_TA* Persona; // 0x0060 (0x0008)
[0x0001000000000000]
class FString ClubTagName; // 0x0068 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
};

```

```
uint64_t ClubID; // 0x0078 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PersonaClubData");
}

return uClassPointer;
};

};

// Class TAGame.__CrowdActorManager_TA__OverrideCrowdActorSounds_0x1
// 0x0008 (0x0060 - 0x0068)
class U__CrowdActorManager_TA__OverrideCrowdActorSounds_0x1 : public UObject
{
public:
class UCrowdSoundsBase_TA* NewSounds; // 0x0060
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CrowdActorManager_TA__OverrideCrowdActorSounds_0x1");
}

return uClassPointer;
};

void __CrowdActorManager_TA__OverrideCrowdActorSounds_0x1(class ACrowdActor_TA* C);
};

// Class TAGame.CrowdSoundsBase_TA
// 0x009C (0x0060 - 0x00FC)
class UCrowdSoundsBase_TA : public UObject
{
public:
class UAkSoundCue* Sound; // 0x0060 (0x0008)
[0x0001000000000001] (CPF_Edit)
struct FCrowdExcitementSounds GameStart; // 0x0068 (0x0028)
[0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds RoundStart; // 0x0090 (0x0028)
[0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
```

```

TArray<struct FCrowdExcitementSounds> MatchEndCountDown; // 0x00B8 (0x0010) [0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UAkSoundCue* LowEnergyChant; // 0x00C8 (0x0008) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* OvertimeStart; // 0x00D0 (0x0008) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* Win; // 0x00D8 (0x0008) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* Lose; // 0x00E0 (0x0008) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* Forfeit; // 0x00E8 (0x0008) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* MatchEndedEncore; // 0x00F0 (0x0008) [0x0001000000000001] (CPF_Edit)
float MatchEndedEncoreDelay; // 0x00F8 (0x0004) [0x0001000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundsBase_TA");
}

return uClassPointer;
};

};

// Class TAGame.CrowdActor_TA
// 0x0070 (0x0268 - 0x02D8)
class ACrowdActor_TA : public AActor
{
public:
class UAkSoundCue* CrowdSound; // 0x0268 (0x0008) [0x0000000000000001] (CPF_Edit)
class UCrowdActorSettings_TA* Settings; // 0x0270 (0x0008) [0x0000000000000001] (CPF_Edit)
class AGameEvent_Soccar_TA* GameEvent; // 0x0278 (0x0008) [0x0000000100002020] (CPF_Net | CPF_Transient)
float TargetIdleNoise; // 0x0280 (0x0004) [0x0000000000002000] (CPF_Transient)
float TargetIdleNoiseRate; // 0x0284 (0x0004) [0x0000000000002000] (CPF_Transient)
float CurrentIdleNoise; // 0x0288 (0x0004) [0x0000000000002000] (CPF_Transient)
float ModifiedNoise; // 0x028C (0x0004) [0x0000000000002020] (CPF_Net | CPF_Transient)
float CurrentNoise; // 0x0290 (0x0004) [0x0000000000002000] (CPF_Transient)
TArray<struct FCrowdNoiseModifierInstance> NoiseModifiers; // 0x0298

```

```

(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FCrowdNoiseModifierInstance           TimeAlmostOutNoiseInstance;          // 0x02A8 (0x0008) [0x0000000000002000] (CPF_Transient)
float                                         CurrentBallGoalDistanceNoise;        // 0x02B0 (0x0004) [0x0000000000002000] (CPF_Transient)
class UAkSoundCue*                          ReplicatedOneShotSound;            // 0x02B8 (0x0008) [0x0000000100002020] (CPF_Net | CPF_Transient)
int32_t                                     ReplicatedCountDownNumber;         // 0x02C0 (0x0004) [0x0000000100002020] (CPF_Net | CPF_Transient)
int32_t                                     ReplicatedRoundCountDownNumber;    // 0x02C4 (0x0004) (0x0004) [0x0000000100002020] (CPF_Net | CPF_Transient)
float                                         LastBallHitNoiseTime;             // 0x02C8 (0x0004) [0x0000000000002000] (CPF_Transient)
int32_t                                     LastBallHitNoiseLevel;            // 0x02CC (0x0004) [0x0000000000002000] (CPF_Transient)
class UCrowdSoundsBase_TA*                  Sounds;                           // 0x02D0 (0x0008) [0x0001000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdActor_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
void HandleCarTouch(class ABall_TA* Ball, class ACar_TA* HitCar, uint8_t HitType);
void HandleBallRemoved(class AGameEvent_Soccar_TA* G, class ABall_TA* Ball);
void HandleBallAdded(class AGameEvent_Soccar_TA* G, class ABall_TA* Ball);
void HandleGaveStat(class AStatFactory_TA* Factory, class APRI_TA* ToPRI, class UStatEvent_TA* StatEvent, class ABall_TA* Ball, int32_t BallHitIndex, class APRI_TA* Victim, class UGoal_TA* Goal);
void HandleCountdownTimeUpdated(class AGameEvent_TA* InGameEvent, int32_t Seconds);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void UpdateTimeAlmostOutNoiseModifier(class AGameEvent_Soccar_TA* InGameEvent);
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleGameStateChanged(class AGameEvent_TA* InGameEvent);
void PlayRoundCountDownSound(int32_t CountDownNumber);
void PlayCountDownSound(int32_t CountDownNumber);
void HandleEndGameCountDown(class AGameEvent_Soccar_TA* G, int32_t TimeRemaining);
void HandleStingerForfeit(class AGameEvent_Soccar_TA* G);
void HandleStingerLose(class AGameEvent_Soccar_TA* G);
void HandleStingerWin(class AGameEvent_Soccar_TA* G);
void eventTick(float DeltaTime);
void TickBallGoalDistance();
void SetNoise(float NewNoise);
void TickNoise(float DeltaTime, float TargetNoise);
void ClearOneShotSound();

```

```

void PlayOneShotSound(class UAkSoundCue* Sound);
void AddNoiseModifier(class UCrowdNoiseModifier_TA* Modifier);
struct FCrowdNoiseModifierInstance CreateNoiseModifierInstance(class
UCrowdNoiseModifier_TA* Modifier);
void TickNoiseModifiers(float DeltaTime);
void TickIdleNoise(float DeltaTime);
void UpdateTargetIdleNoise();
void Init(class AGameEvent_Soccar_TA* InGameEvent);
void eventDestroyed();
void eventPostBeginPlay();
float GetRandomValue(struct FRandomRange Range);
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.CrowdActorManager_TA
// 0x0018 (0x0268 - 0x0280)
class ACrowdActorManager_TA : public AActor
{
public:
TArray<class ACrowdActor_TA*> CrowdActors; // 0x0268
(0x0010) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UCrowdSoundsBase_TA* Sounds; // 0x0278 (0x0008)
[0x0001004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdActorManager_TA");
}

return uClassPointer;
};

bool __CrowdActorManager_TA__PlayCrowdSound_0x1(class ACrowdActor_TA* C);
bool __CrowdActorManager_TA__StopCrowdSound_0x1(class ACrowdActor_TA* C);
void __CrowdActorManager_TA__Destroyed_0x1(class ACrowdActor_TA* C);
void eventDestroyed();
void StopCrowdExcitementSound(class UAkSoundCue* Sound, class AActor* FromActor);
void StopCrowdSound(struct FScriptDelegate GetSoundDelegate);
void PlayCrowdActorSound(class UAkSoundCue* Sound, class AActor* FromActor);
void PlayCrowdSound(struct FScriptDelegate GetSoundDelegate);
void OverrideCrowdActorSounds(class UCrowdSoundsBase_TA* NewSounds);
void SetCrowdSounds(class UCrowdSoundsBase_TA* NewSounds);
void SetCrowdActors(TArray<class ACrowdActor_TA*> Actors);
};

// Class TAGame.CrowdSoundManagerBase_TA
// 0x0094 (0x00A4 - 0x0138)
class UCrowdSoundManagerBase_TA : public UActorComponent_X
{

```

```

public:
class UCrowdSoundSettingsBase_TA*           BaseSettings;          // 0x00A8
(0x0008) [0x0001000000000000]
class ACrowdActorManager_TA*                CrowdActorManager;    // 0x00B0
(0x0008) [0x0001000000000000]
class UCrowdSoundsBase_TA*                 Sounds;              // 0x00B8 (0x0008)
[0x0001004000002000] (CPF_Transient)
unsigned long                  bInitializedMods : 1;      // 0x00C0 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)
unsigned long                  bIn0Seconds : 1;        // 0x00C0 (0x0004)
[0x0001000000002000] [0x00000002] (CPF_Transient)
unsigned long                  bLowEnergy : 1;        // 0x00C0 (0x0004)
[0x0001000000002000] [0x00000004] (CPF_Transient)
class FString                   DebugLastExcitementSoundValue; // 0x00C8
(0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
float                         DebugLastExcitementSoundTime; // 0x00D8 (0x0004)
[0x0001000000002000] (CPF_Transient)
float                         TimeSince0Seconds;       // 0x00DC (0x0004)
[0x0001000000002000] (CPF_Transient)
class UDynamicValueModifier_X*     Mod0Seconds;          // 0x00E0
(0x0008) [0x000100000002000] (CPF_Transient)
float                         CurrentLowEnergyDuration; // 0x00E8 (0x0004)
[0x0001000000002000] (CPF_Transient)
float                         ChantLowEnergyDuration; // 0x00EC (0x0004)
[0x0001000000002000] (CPF_Transient)
struct FName                   LastExcitementState;    // 0x00F0 (0x0008)
[0x0001000000002000] (CPF_Transient)
struct FName                   LastTensionState;      // 0x00F8 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString                   AprilCrowdOverridePath; // 0x0100 (0x0010)
[0x0001080000400002] (CPF_Const | CPF_NeedCtorLink)
struct FVector                 AprilCrowdLocation;    // 0x0110 (0x000C)
[0x0001000000000000]
struct FScriptDelegate         __GetSound__Delegate;   // 0x0120 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundManagerBase_TA");
}

return uClassPointer;
};

class UAkSoundCue* __CrowdSoundManagerBase_TA__InitCrowd_0x1(class
UCrowdSoundsBase_TA* S);
float __CrowdSoundManagerBase_TA__InitModifiers_0x2();
float __CrowdSoundManagerBase_TA__InitModifiers_0x1();
void __CrowdSoundManagerBase_TA__HandleGameStateChanged_0x2(class ACrowdActor_TA*

```

```
C);
void __CrowdSoundManagerBase_TA__HandleGameStateChanged_0x1(class ACrowdActor_TA*
C);
class UAkSoundCue* __CrowdSoundManagerBase_TA__HandleOvertimeUpdated_0x1(class
UCrowdSoundsBase_TA* S);
class UAkSoundCue* __CrowdSoundManagerBase_TA__PlayMatchEndedEncore_0x1(class
UCrowdSoundsBase_TA* S);
class UAkSoundCue* __CrowdSoundManagerBase_TA__HandleGameWinnerSet_0x3(class
UCrowdSoundsBase_TA* S);
class UAkSoundCue* __CrowdSoundManagerBase_TA__HandleGameWinnerSet_0x2(class
UCrowdSoundsBase_TA* S);
class UAkSoundCue* __CrowdSoundManagerBase_TA__HandleGameWinnerSet_0x1(class
UCrowdSoundsBase_TA* S);
void __CrowdSoundManagerBase_TA__HandleGameTimeUpdated_0x1(class ACrowdActor_TA*
C);
class UAkSoundCue* __CrowdSoundManagerBase_TA__CheckEnergy_0x1(class
UCrowdSoundsBase_TA* S);
void PrintDebugInfo(class UDebugDrawer* Drawer);
float GetCrowdValue(uint8_t Type);
class UCrowdSoundSettingsBase_TA* GetBaseCrowSettings();
void StopCrowdExcitementSound(struct FCrowdExcitementSounds ExcitementSounds, class
AActor* FromActor);
void PlayCrowdExcitementSoundNoCooldown(struct FCrowdExcitementSounds
ExcitementSounds, class AActor* FromActor);
void PlayCrowdExcitementSound(class AActor* FromActor, struct FCrowdExcitementSounds&
ExcitementSounds);
void SetNextChantDuration();
void CheckEnergy(float DeltaTime);
void Update0Seconds(float DeltaTime);
void PlayMatchEndCountdown(int32_t InSecondsRemaining, class UCrowdSoundsBase_TA*
InSounds, class AActor* FromActor);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleGameWinnerSet(class AGameEvent_Soccar_TA* InGameEvent);
float GetTimeRemaining();
void UpdateStates();
void TickValues(float DeltaTime);
void TickActive(float DeltaTime);
void eventTick(float DeltaTime);
void PlayMatchEndedEncore();
void HandleMatchEnded(class AGameEvent_Soccar_TA* G);
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleGameStateChanged(class AGameEvent_TA* G);
void ClearPendingCrowdFlags();
void StopCrowdActorSound(struct FScriptDelegate GetSoundDelegate);
void PlayCrowdActorSound(struct FScriptDelegate GetSoundDelegate);
class UDynamicValueModifier_X* AddDynamicModifier(class UDynamicValue_X* Value, class
UDynamicValueModifier_X* ModArchetype);
bool CanModifyCrowd();
class UDynamicValueModifier_X* CreateCurveModifier(class UDynamicValueModifierCurve_X*
ModArchetype, struct FScriptDelegate InDelegate);
void RemoveAllModifiers();
void InitModifiers();
void TryInitModifiers();
void InitGameObserver();
```

```

void InitCrowd();
void eventDetached();
void TickInit();
void eventAttached();
class UAkSoundCue* GetSound(class UCrowdSoundsBase_TA* InSounds);
};

// Class TAGame.CrowdSoundSettingsBase_TA
// 0x00A0 (0x0060 - 0x0100)
class UCrowdSoundSettingsBase_TA : public UObject
{
public:
class UDYNAMICVALUE_X* Tension; // 0x0060 (0x0008)
[0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
class UDYNAMICVALUE_X* Excitement; // 0x0068 (0x0008)
[0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
class UDYNAMICVALUE_X* Upset; // 0x0070 (0x0008)
[0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
class UDYNAMICVALUE_X* Anticipation; // 0x0078 (0x0008)
[0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
class UDYNAMICVALUEMODIFIERDURATION_X* ExcitementGameStart; // 0x0080
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICVALUEMODIFIERDURATION_X* ExcitementRoundStart; // 0x0088
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICVALUEMODIFIERDURATION_X* ExcitementOvertimeStart; // 0x0090
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICVALUEMODIFIERCURVE_X* ExcitementTimeRemaining; // 0x0098
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICVALUEMODIFIERCURVE_X* Excitement0Seconds; // 0x00A0
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICVALUEMODIFIERCURVE_X* Anticipation0Seconds; // 0x00A8
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICVALUEMODIFIERCURVE_X* AnticipationExcitementMultiplier; // 0x00B0
(0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
struct FCrowdDefense DefenseChantSettings; // 0x00B8 (0x000C)
[0x0001000000000001] (CPF_Edit)
float LowEnergyMaxExcitement; // 0x00C4 (0x0004)
[0x0001000000000001] (CPF_Edit)
float LowEnergyMaxTension; // 0x00C8 (0x0004)
[0x0001000000000001] (CPF_Edit)
float ChantLowEnergyDuration; // 0x00CC (0x0004)
[0x0001000000000001] (CPF_Edit)
float ChantLowEnergyRandomDuration; // 0x00D0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float KickoffTimeRemaining; // 0x00D4 (0x0004)
[0x0001000000000001] (CPF_Edit)
float PostGameExcitement; // 0x00D8 (0x0004)

```

```
[0x0001000000000001] (CPF_Edit)
TArray<struct FCrowdValueState>           ExcitementStates;          // 0x00E0
(0x0010) [0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FCrowdValueState>           TensionStates;          // 0x00F0
(0x0010) [0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundSettingsBase_TA");
}

return uClassPointer;
};

// Class TAGame.__CrowdActorManager_TA__PlayCrowdSound_0x2
// 0x0018 (0x0060 - 0x0078)
class U__CrowdActorManager_TA__PlayCrowdSound_0x2 : public UObject
{
public:
struct FScriptDelegate           GetSoundDelegate;          // 0x0060 (0x0018)
[0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CrowdActorManager_TA__PlayCrowdSound_0x2");
}

return uClassPointer;
};

void __CrowdActorManager_TA__PlayCrowdSound_0x2(class ACrowdActor_TA* C);
};

// Class TAGame.__CrowdActorManager_TA__StopCrowdSound_0x2
// 0x0018 (0x0060 - 0x0078)
class U__CrowdActorManager_TA__StopCrowdSound_0x2 : public UObject
{
public:
struct FScriptDelegate           GetSoundDelegate;          // 0x0060 (0x0018)
[0x000100000400000] (CPF_NeedCtorLink)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._CrowdActorManager_TA__StopCrowdSound_0x2");
}

return uClassPointer;
};

void __CrowdActorManager_TA__StopCrowdSound_0x2(class ACrowdActor_TA* C);
};

// Class TAGame._CrumbTrails_TA__GetActiveCrumbs_0x1
// 0x0010 (0x0060 - 0x0070)
class U_CrumbTrails_TA__GetActiveCrumbs_0x1 : public UObject
{
public:
TArray<struct FName> Crumbs; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._CrumbTrails_TA__GetActiveCrumbs_0x1");
}

return uClassPointer;
};

void __CrumbTrails_TA__GetActiveCrumbs_0x1(struct FName C);
};

// Class TAGame.CrumbTrails_TA
// 0x0058 (0x0060 - 0x00B8)
class UCrumbTrails_TA : public UObject
{
public:
TArray<struct FCrumbTrailData> CrumbTrails; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class UCrumbTrailRedefinition_TA* CrumbRedefine; // 0x0070
(0x0008) [0x0001800000000000]
class UCrumbTrailSave_TA* CrumbSave; // 0x0078 (0x0008)
[0x0001000000000000]
unsigned long bTrailSetup : 1; // 0x0080 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)

```

```
struct FScriptDelegate           __EventActiveTrailsChanged__Delegate;      // 0x0088
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventActiveTrailCompleted__Delegate;     // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrumbTrails_TA");
}

return uClassPointer;
};

void DebugPrintActiveCrumbs();
void DebugPrintAll();
void HandleTrailComplete(int32_t Index);
void GetCrumbs(struct FName Crumb TrailName, TArray<struct FName>& OutCrumbs);
void GetActiveCrumbs(TArray<struct FName>& OutCrumbs);
void ActivateTrailByName(struct FName TrailName);
bool CanActivateTrailByIndex(int32_t Index);
bool CanActivateTrail(struct FName TrailName);
bool ActivateTrail(int32_t Index);
void SetupCompletionCondition(int32_t Index);
void SetupActivationCondition(int32_t Index);
void SetupTrails();
void HandleSaveAdded(class UCrumbTrailSave_TA* InCrumbSave);
void HandleRedefinitionChange();
class UCrumbTrailMetrics_TA* GetMetrics();
void Init(class ULocalPlayer_TA* LP);
void EventActiveTrailCompleted(struct FName Crumb Trail);
void EventActiveTrailsChanged(class UCrumbTrails_TA* Crumb TrailMgr);
};

// Class TAGame.__CrumbTrails_TA__HandleTrailComplete_0x1
// 0x0004 (0x0060 - 0x0064)
class U__CrumbTrails_TA__HandleTrailComplete_0x1 : public UObject
{
public:
int32_t                      Index;                                // 0x0060 (0x0004)
[0x0010000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
```

```

TAGame.__CrumbTrails_TA__HandleTrailComplete_0x1");
}

return uClassPointer;
};

bool __CrumbTrails_TA__HandleTrailComplete_0x1(struct FName T);
};

// Class TAGame.__CrumbTrails_TA__SetupActivationCondition_0x1
// 0x0004 (0x0060 - 0x0064)
class U__CrumbTrails_TA__SetupActivationCondition_0x1 : public UObject
{
public:
int32_t Index; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__CrumbTrails_TA__SetupActivationCondition_0x1");
}

return uClassPointer;
};

void __CrumbTrails_TA__SetupActivationCondition_0x2(class UTriggerCondition_TA* _);
void __CrumbTrails_TA__SetupActivationCondition_0x1(class UTriggerClump_TA* C);
};

// Class TAGame.TriggerClump_TA
// 0x0010 (0x0060 - 0x0070)
class UTriggerClump_TA : public UObject
{
public:
TArray<class UStatusTrigger_X*> Triggers; // 0x0060 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TriggerClump_TA");
}

return uClassPointer;
};

```

```

};

void DebugPrint();
};

// Class TAGame.TriggerCondition_TA
// 0x0048 (0x0060 - 0x00A8)
class UTriggerCondition_TA : public UObject
{
public:
TArray<class UTriggerClump_TA*> Clumps; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate Callback; // 0x0070 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bRepeatable : 1; // 0x0088 (0x0004)
[0x0000000000000000] [0x00000001]
struct FScriptDelegate __OnTriggersMet__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TriggerCondition_TA");
}

return uClassPointer;
};

bool __TriggerCondition_TA__HasTriggerConditionBeenMet_0x1(class UTriggerClump_TA* G);
bool __TriggerCondition_TA__HasTriggerConditionBeenMet_0x2(class UStatusTrigger_X* T);
void DebugPrint();
bool HasTriggerConditionBeenMet();
void Unregister();
void Register();
class UTriggerCondition_TA* SetCallback(struct FScriptDelegate InCallback);
class UTriggerCondition_TA* SetRepeatable(unsigned long blsRepeatable);
class UTriggerCondition_TA* AddTriggers(TArray<class UStatusTrigger_X*>& InTriggers);
class UTriggerCondition_TA* AddClump(class UTriggerClump_TA*& InClump);
void OnTriggersMet(class UTriggerCondition_TA* InTriggerCondition);
};

// Class TAGame._CrumbTrails_TA_SetupCompletionCondition_0x1
// 0x0004 (0x0060 - 0x0064)
class U_CrumbTrails_TA_SetupCompletionCondition_0x1 : public UObject
{
public:
int32_t Index; // 0x0060 (0x0004)
[0x0001000000000000]

public:

```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._CrumbTrails_TA_SetupCompletionCondition_0x1");
}

return uClassPointer;
};

void __CrumbTrails_TA_SetupCompletionCondition_0x1(class UTriggerCondition_TA* C);
};

// Class TAGame._CrumbTrails_TA_SetupTrails_0x1
// 0x0004 (0x0060 - 0x0064)
class U_CrumbTrails_TA_SetupTrails_0x1 : public UObject
{
public:
int32_t Index; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._CrumbTrails_TA_SetupTrails_0x1");
}

return uClassPointer;
};

void __CrumbTrails_TA_SetupTrails_0x1(class UTriggerClump_TA* X);
};

// Class TAGame._CurrencyProductSet_TA_GetCurrencyIndexFromHashID_0x1
// 0x0004 (0x0060 - 0x0064)
class U_CurrencyProductSet_TA_GetCurrencyIndexFromHashID_0x1 : public UObject
{
public:
struct FProductHashID CurrencyHashID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
```

```

{
uClassPointer = UObject::FindClass("Class
TAGame.__CurrencyProductSet_TA__GetCurrencyIndexFromHashID_0x1");
}

return uClassPointer;
};

bool __CurrencyProductSet_TA__GetCurrencyIndexFromHashID_0x1(struct FCurrency C);
};

// Class TAGame.CurrencyProductSet_TA
// 0x0018 (0x0060 - 0x0078)
class UCurrencyProductSet_TA : public UObject
{
public:
class UWallet_TA*           PlayerWallet;           // 0x0060 (0x0008)
[0x0000000000002000] (CPF_Transient)
TArray<struct FCurrencyProduct>   CurrencyProducts;    // 0x0068
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CurrencyProductSet_TA");
}
}

return uClassPointer;
};

void __CurrencyProductSet_TA__Init_0x1(class UWallet_TA* InWallet);
bool __CurrencyProductSet_TA__GetCurrenciesWithBalance_0x1(struct FCurrency C);
struct FCurrency __CurrencyProductSet_TA__GetCurrencies_0x1(struct FCurrencyProduct C);
int32_t GetCurrencyIndexFromHashID(struct FProductHashID CurrencyHashID);
void Clear();
TArray<struct FCurrency> GetCurrencies();
TArray<struct FCurrency> GetCurrenciesWithBalance();
void SetQuantity(struct FProductHashID CurrencyHashID, int32_t Quantity);
int32_t GetQuantity(struct FProductHashID CurrencyHashID);
void Init(class UObjectProvider* Owner);
};

// Class TAGame.__EngagementEventsConfig_TA__GetNextEventTime_0x1
// 0x0008 (0x0060 - 0x0068)
class U_EngagementEventsConfig_TA__GetNextEventTime_0x1 : public UObject
{
public:
uint64_t           CurrentTime;           // 0x0060 (0x0008)
[0x0000000000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EngagementEventsConfig_TA__GetNextEventTime_0x1");
}

return uClassPointer;
};

bool __EngagementEventsConfig_TA__GetNextEventTime_0x1(uint64_t Time);
};

// Class TAGame.EngagementEventsConfig_TA
// 0x0080 (0x0078 - 0x00F8)
class UEngagementEventsConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bEngagementEventIconsEnabled : 1;      // 0x0078
(0x0004) [0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
TArray<struct FEngagementEventData>    Events;           // 0x0080
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FEngagementEventStatus>   EventStatuses;     // 0x0090
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FEngagementEventStatus        CurrentEventStatus; // 0x0A0
(0x0030) [0x0000000000000000]
float                           PopUpDuration;       // 0x0D0 (0x0004)
[0x0000000000000000]
uint64_t                         SoonInSeconds;      // 0xD8 (0x0008)
[0x0000000000000000]
struct FScriptDelegate           __EngagementEventSort__Delegate; // 0xE0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngagementEventsConfig_TA");
}

return uClassPointer;
};

void GetEventTimeRemaining();
static uint64_t GetNextEventTime(uint64_t CurrentTime, TArray<struct
FEngagementEventData>& InEvents);
static TArray<uint64_t> EventToEpochTimes(struct FEngagementEventData Event);

```

```
void UpdateEvents(uint64_t CurrentTime);
void UpdateEventsCurrentTime();
void Apply();
void RemoveOverlappingEvents();
int32_t EngagementEventSort(struct FEngagementEventData A, struct FEngagementEventData B);
void eventConstruct();
};

// Class TAGame.__EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x5
// 0x0010 (0x0060 - 0x0070)
class U__EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x5 : public UObject
{
public:
TArray<class UProduct_TA*> UnlockedBodies; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x5");
}

return uClassPointer;
};

bool __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x5(struct
FBodyToEngineAudioPair Pair);
};

// Class TAGame.EngineAudioUnlocker_TA
// 0x0000 (0x0060 - 0x0060)
class UEngineAudioUnlocker_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioUnlocker_TA");
}

return uClassPointer;
};
```

```

static int32_t __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x8(class UProduct_TA*
P);
static int32_t __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x7(class UProduct_TA*
P);
static class UProduct_TA* __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x6(struct
FBodyToEngineAudioPair Pair);
static bool __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x4(class UProduct_TA*
Product);
static bool __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x3(class UProduct_TA*
Product);
static bool __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x2(class UProduct_TA*
Product);
static int32_t __EngineAudioUnlocker_TA__FixupUnlockedEngineAudio_0x1(class
UOnlineProduct_TA* Product);
static void FixupUnlockedEngineAudio(class USaveData_TA* SaveData);
};

// Class TAGame.EOSGameClipsMetrics_TA
// 0x0038 (0x0080 - 0x00B8)
class UEOSGameClipsMetrics_TA : public UMetricsGroup_X
{
public:
    class UEOSGameClipsManager_TA*           GameClipsManager;          // 0x0080
    (0x0008) [0x0001804000000000]
    class UOnlineGameClipsInterface*         GameClipsInterface_Object; // 0x0088
    (0x0008) [0x0001004000000000]
    class UOnlineGameClipsInterface*         GameClipsInterface_Interface; // 0x0090
    (0x0008) [0x0001004000000000]
    TArray<struct FEOSGameClipsClipInfo>     ClipsUploading;           // 0x0098
    (0x0010) [0x0001004000400000] (CPF_NeedCtorLink)
    TArray<struct FName>                   GeneralErrors;            // 0x00A8 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.EOSGameClipsMetrics_TA");
        }

        return uClassPointer;
    };

    void RecordClipInfo(struct FEOSGameClipsClipInfo InClipInfoMetric);
    void RecordAccountLinking(struct FEOSGameClipsAccountLinkInfo InAccountMetric);
    int32_t AddClip(class FString InEpicAccountId, int32_t InClipId);
    void HandleCreateClipInitiated(class FString InEpicAccountId, int32_t InClipId, class FString&
InClipType);
    void HandleClipStatusChanged(class FString InEpicAccountId, int32_t InClipId, uint8_t
InNewClipStatus);
    void HandleClipErrorOccurred(class FString InEpicAccountId, int32_t InClipId, class UErrorType*

```

```

InErrorType);
void HandleGeneralErrorOccurred(class UErrorType* InErrorType);
void HandleConnectionStatusChanged(class FString InEpicAccountId, uint8_t InConnection,
uint8_t InNewConnectionStatus);
void HandleGameClipsManagerChanged();
void SetGameClipsInterface(class UOnlineGameClipsInterface* InGameClipsInterface);
};

// Class TAGame.__EOSGameClipsMetrics_TA__RecordClipInfo_0x1
// 0x0040 (0x0060 - 0x00A0)
class U__EOSGameClipsMetrics_TA__RecordClipInfo_0x1 : public UObject
{
public:
struct FEOSGameClipsClipInfo           InClipInfoMetric;          // 0x0060 (0x0040)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSGameClipsMetrics_TA__RecordClipInfo_0x1");
}

return uClassPointer;
};

void __EOSGameClipsMetrics_TA__RecordClipInfo_0x1(struct FName ErrorName);
};

// Class TAGame.__EOSVoiceManager_TA__ClearRoomCredentialsForPlayer_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__EOSVoiceManager_TA__ClearRoomCredentialsForPlayer_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;                      // 0x0060 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__ClearRoomCredentialsForPlayer_0x1");
}

return uClassPointer;
};

```

```

bool __EOSVoiceManager_TA__ClearRoomCredentialsForPlayer_0x1(struct FVoiceRoomToken
P);
};

// Class TAGame.EOSVoiceManager_TA
// 0x0150 (0x0060 - 0x01B0)
class UEOSVoiceManager_TA : public UObject
{
public:
class UOnlineVoiceInterfaceEOS*           VoiceInterface;           // 0x0060 (0x0008)
[0x0001004000000000]
TArray<struct FVoiceRoom>               VoiceRooms;             // 0x0068 (0x0010)
[0x0001008000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FPendingRoomJoin>          PendingRoomJoins;       // 0x0078
(0x0010) [0x0001008000400000] (CPF_NeedCtorLink)
class UEOSVoiceConfig_TA*                VoiceConfig;            // 0x0088 (0x0008)
[0x0001800000000000]
class UOnlineGame_TA*                   OnlineGame;             // 0x0090 (0x0008)
[0x0001800000000000]
class UOnlineGameParty_TA*              OnlineGameParty;        // 0x0098
(0x0008) [0x0001800000000000]
class UPersonas_TA*                    Personas;               // 0x00A0 (0x0008)
[0x0001800000000000]
int32_t                                NoTeamNum;              // 0x00A8 (0x0004)
[0x0001000000000002] (CPF_Const)
int32_t                                SpectatorTeamNum;        // 0x00AC (0x0004)
[0x0001004000000002] (CPF_Const)
struct FUniqueNetId                     PrimaryPlayerId;        // 0x00B0 (0x0048)
[0x000100000400000] (CPF_NeedCtorLink)
TArray<class UOnlinePlayer_X*>          PlayersRequestingPartyTokens; // 0x00F8
(0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate                  __EventPlayerVoiceErrorUpdated__Delegate; // 0x0108
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                  __EventTrackedVoiceRoomChanged__Delegate; // 0x0120
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                  __EventJoinedVoiceRoom__Delegate;      // 0x0138
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                  __EventLeftVoiceRoom__Delegate;       // 0x0150
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                  __EventRemotePlayerJoinedVoiceRoom__Delegate; // 0x0168
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                  __EventRemotePlayerLeftVoiceRoom__Delegate; // 0x0180
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                  __EventPlayerStatusChange__Delegate;    // 0x0198
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.EOSVoiceManager_TA");
}

return uClassPointer;
};

void __EOSVoiceManager_TA__Construct_0x1();
void __EOSVoiceManager_TA__Construct_0x2(class AGameEvent_TA* _);
bool __EOSVoiceManager_TA__HandleGameEventDestroyed_0x2(struct FVoiceRoom P);
bool __EOSVoiceManager_TA__HandleGameEventDestroyed_0x1(struct FVoiceRoom V);
bool __EOSVoiceManager_TA__GetPossibleVoiceMemberIds_0x4(struct FUniqueNetId P);
struct FUniqueNetId __EOSVoiceManager_TA__GetPossibleVoiceMemberIds_0x3(class
APlayerReplicationInfo* P);
bool __EOSVoiceManager_TA__GetPossibleVoiceMemberIds_0x2(class APlayerReplicationInfo*
P);
struct FUniqueNetId __EOSVoiceManager_TA__GetPossibleVoiceMemberIds_0x1(struct
FPartyMember P);
bool __EOSVoiceManager_TA__HandlePartyDestroyed_0x1(struct FVoiceRoom P);
static bool IsSpectatorTeamNum(int32_t TeamNum);
static bool IsVoiceAllowedForTeam(int32_t TeamNum, class AGameEvent_Soccar_TA*
GameEvent);
static class FString GenerateMatchRoomName(class FString MatchGuid, int32_t TeamNum);
class UEOSVoiceMetrics_TA* GetMetrics();
class FString GetCurrentMatchRoomName(int32_t TeamNum);
class UOnlinePlayer_X* GetOnlinePlayerFromEpicId(class FString EpicAccountId);
class FString GetEpicAccountId(class UOnlinePlayer_X* OnlinePlayer);
struct FVoiceRoom GetPlayerVoiceRoomByEpicId(class FString EpicAccountId);
struct FVoiceRoom GetPlayerVoiceRoom(struct FUniqueNetId PlayerID);
void GetMemberIndex(class FString EpicAccountId, class FString RoomName, int32_t&
RoomIndex, int32_t& MemberIndex);
TArray<class UOnlinePlayer_X*> GetLocalPlayersInRoom(class FString RoomName);
bool IsLocalPlayerInRoom(class FString RoomName);
struct FVoiceRoom GetVoiceRoomByName(class FString RoomName);
void HandleRemovePartyMember(class UOnlineGameParty_X* Party, struct FUniqueNetId
PlayerID);
void HandlePartyMemberChangedRooms(class UOnlineMessageComponent_X* Component,
class UPartyMessage_CurrentVoiceRoom_TA* Message);
void HandlePartyDestroyed(class UOnlineGameParty_X* Party);
void HandlePartyCreated(class UOnlineGameParty_X* Party, unsigned long bWasSuccessful);
static void OnJoinParty(class UEOSVoiceManager_TA* VoiceManager, class
UPartyJoinedEvent_X* );
uint8_t GetCurrentRoomTypeForPlayer(class UOnlinePlayer_X* OnlinePlayer);
class FString GetCurrentRoomNameForPlayer(class UOnlinePlayer_X* OnlinePlayer);
void ClearRoomCredentialsForPlayer(struct FUniqueNetId PlayerID, class FString RoomName);
bool HasRoomCredentials(struct FUniqueNetId PlayerID, class FString RoomName);
void SetPlayerBlockStatus(class UOnlinePlayer_X* OnlinePlayer, class FString EpicAccountId,
class FString RoomName, unsigned long bBlock);
void SetPlayerMuteStatus(class UOnlinePlayer_X* OnlinePlayer, class FString EpicAccountId,
class FString RoomName, unsigned long bMute);
void SetLocalMuteStatus(class UOnlinePlayer_X* OnlinePlayer, unsigned long bMuted);
void AlertVoiceError(class UOnlinePlayer_X* OnlinePlayer, class UError* VoiceError);
void AlertMatchRoomTokenRequestFailed(class UOnlinePlayer_X* OnlinePlayer, class UError*
VoiceError);
void AlertVoiceBanned(class UOnlinePlayer_X* OnlinePlayer);

```

```
class UEOSVoiceSettingsSave_TA* GetPlayerVoiceSettingsSave(class UOnlinePlayer_X* OnlinePlayer);
bool IsPlayerVoiceSettingEnabled(class UOnlinePlayer_X* OnlinePlayer);
uint8_t GetMaxChatPermissionLevel();
class UError* GetPlayerPermissionError(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName);
class UError* GetJoinVoiceRoomError(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName);
bool HasJoinVoiceRoomError(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName);
bool IsPlayerInVoiceRoom(class UOnlinePlayer_X* OnlinePlayer);
uint8_t GetUserState(class UOnlinePlayer_X* OnlinePlayer);
void LeaveRoom(class UOnlinePlayer_X* OnlinePlayer);
bool RemovePlayerFromRoom(class FString EpicAccountId, class FString RoomName);
TArray<struct FUniqueNetId> GetPossibleVoiceMemberIds();
void HandlePlayerStatusChanged(class FString EpicAccountId, class FString RoomName, struct FVoiceRoomMemberStatus MemberStatus);
void HandleRemotePlayerLeftVoiceRoom(class FString EpicAccountId, class FString RoomName);
void HandleRequestedLinkedAccounts(unsigned long bSuccess, TArray<struct FLinkedAccountData> LinkedData);
class FString GetPersonaEpicId(struct FUniqueNetId PlayerID);
void HandleRemotePlayerJoinedVoiceRoom(class FString EpicAccountId, class FString RoomName);
void HandleRemovedFromVoiceRoom(class FString EpicAccountId, class FString RoomName, uint8_t RemoveReason);
void HandleLeftVoiceRoom(class FString EpicAccountId, class FString RoomName, uint8_t LeaveCode);
static class UError* GetResultCodeError(uint8_t ResultCode);
void HandleJoinedVoiceRoom(class FString EpicAccountId, class FString RoomName, uint8_t JoinResult);
void JoinRoom(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName);
void HandlePartyVoiceToken(class URPC_RequestPartyVoiceToken_TA* RPC);
void SendPartyVoiceTokenRequest(class UOnlinePlayer_X* OnlinePlayer);
void RequestPartyRoomTokens();
void ClearTrackedVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, uint8_t RoomType);
void SetTrackedVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName);
bool ShouldJoinVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName);
static bool RoomCredentialsAreValid(struct FVoiceRoomCredentials RoomCredentials);
class UError* AddVoiceRoomCredentials(struct FUniqueNetId PlayerID, struct FVoiceRoomCredentials RoomCredentials, uint8_t RoomType);
void AddVoiceRoom(class FString RoomName, uint8_t RoomType);
void JoinRoomAllUsers(class FString RoomName);
void HandlePlayerMatchRoomChanged(class APRI_TA* PRI);
void HandlePlayerRemovedFromMatch(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void HandleGameEventDestroyed();
void UpdateFeatureEnabled();
void HandleVoiceConfig();
TArray<struct FVoiceAudioDevice> GetOutputAudioDevices();
TArray<struct FVoiceAudioDevice> GetInputAudioDevices();
bool SetAudioOutputDevice(class UOnlinePlayer_X* OnlinePlayer, class FString DeviceID, float Volume);
bool SetAudioInputDevice(class UOnlinePlayer_X* OnlinePlayer, class FString DeviceID, float Volume);
```

```

static bool IsValidDeviceId(class FString DeviceID, TArray<struct FVoiceAudioDevice>&
AudioDevices);
void SetLocalPlayerRegisteredStatus(unsigned long bRegister, struct FUniqueNetId& PlayerID);
void HandlePrimaryPlayerChanged(class UOnlineGameAccount_X* Account, struct FUniqueNetId
PlayerID);
void eventConstruct();
void EventPlayerStatusChange(class FString EpicAccountId, class FString RoomName, struct
FVoiceRoomMemberStatus MemberStatus);
void EventRemotePlayerLeftVoiceRoom(class FString EpicAccountId, class FString RoomName);
void EventRemotePlayerJoinedVoiceRoom(struct FMappedRoomMember RoomMember, class
FString RoomName);
void EventLeftVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName,
unsigned long bSuccess);
void EventJoinedVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName,
unsigned long bSuccess);
void EventTrackedVoiceRoomChanged(class UOnlinePlayer_X* OnlinePlayer, struct FVoiceRoom
VoiceRoomData);
void EventPlayerVoiceErrorUpdated(class UOnlinePlayer_X* OnlinePlayer, class UError*
VoiceError);
};

// Class TAGame.__EOSVoiceManager_TA__GetPlayerPermissionError_0x1
// 0x0009 (0x0060 - 0x0069)
class U__EOSVoiceManager_TA__GetPlayerPermissionError_0x1 : public UObject
{
public:
class UOnlinePlayer_X*           OnlinePlayer;          // 0x0060 (0x0008)
[0x0001000000000000]
uint8_t                          VoiceChatFilter;    // 0x0068 (0x0001)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__GetPlayerPermissionError_0x1");
}

return uClassPointer;
};

bool __EOSVoiceManager_TA__GetPlayerPermissionError_0x3(struct FMappedRoomMember P);
bool __EOSVoiceManager_TA__GetPlayerPermissionError_0x2(struct FMappedRoomMember P);
bool __EOSVoiceManager_TA__GetPlayerPermissionError_0x1(struct FMappedRoomMember P);
};

// Class TAGame.OnlineGame_TA
// 0x0028 (0x02A0 - 0x02C8)
class UOnlineGame_TA : public UOnlineGame_X
{

```

```

public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x02A0
(0x0008) [0x0001000000000001] (CPF_Edit)
class UOnlineGameTourMatchmaking_TA*        TourMatchmaking;    //
0x02A8 (0x0008) [0x0001000000000001] (CPF_Edit)
class UPersonas_TA*                      Personas;           // 0x02B0 (0x0008)
[0x0000000000000000]
class UClubPersonas_TA*                   ClubPersonas;      // 0x02B8 (0x0008)
[0x0001000000000000]
class UMenuTreeBuilder_TA*                MenuTreeBuilder;   // 0x02C0
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGame_TA");
}

return uClassPointer;
};

void __OnlineGame_TA__OnInit_0x1();
bool CanChatWithPlayer(uint8_t ChatFilter, class UOnlinePlayer_X*& OnlinePlayer, struct FUniqueNetId& OtherPlayer);
bool CanPlayerSeeChatMessage(struct FGFxChatMessage& GFxMessage, class UOnlinePlayer_X*& OnlinePlayer);
bool CanSeeChatMessage(struct FGFxChatMessage& GFxMessage);
void OnMainMenuOpened();
void OnInit();
};

// Class TAGame.__EOSVoiceManager_TA__GetPlayerVoiceRoom_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__EOSVoiceManager_TA__GetPlayerVoiceRoom_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;           // 0x0060 (0x0048)
[0x0001000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__GetPlayerVoiceRoom_0x1");
}
}

```

```

return uClassPointer;
};

bool __EOSVoiceManager_TA__GetPlayerVoiceRoom_0x1(struct FVoiceRoom P);
};

// Class TAGame.__EOSVoiceManager_TA__GetPlayerVoiceRoomByEpicId_0x1
// 0x0010 (0x0060 - 0x0070)
class U__EOSVoiceManager_TA__GetPlayerVoiceRoomByEpicId_0x1 : public UObject
{
public:
class FString EpicAccountId; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__GetPlayerVoiceRoomByEpicId_0x1");
}

return uClassPointer;
};

bool __EOSVoiceManager_TA__GetPlayerVoiceRoomByEpicId_0x1(struct FVoiceRoom P);
};

// Class TAGame.__EOSVoiceManager_TA__GetVoiceRoomByName_0x1
// 0x0010 (0x0060 - 0x0070)
class U__EOSVoiceManager_TA__GetVoiceRoomByName_0x1 : public UObject
{
public:
class FString RoomName; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__GetVoiceRoomByName_0x1");
}

return uClassPointer;
};

bool __EOSVoiceManager_TA__GetVoiceRoomByName_0x1(struct FVoiceRoom P);

```

```
};

// Class TAGame.__EOSVoiceManager_TA__HandlePlayerMatchRoomChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U__EOSVoiceManager_TA__HandlePlayerMatchRoomChanged_0x1 : public UObject
{
public:
class APRI_TA*          PRI;           // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__HandlePlayerMatchRoomChanged_0x1");
}

return uClassPointer;
};

void __EOSVoiceManager_TA__HandlePlayerMatchRoomChanged_0x1(class UPersona_TA* _);

// Class TAGame.__EOSVoiceManager_TA__HandleRemotePlayerJoinedVoiceRoom_0x1
// 0x0020 (0x0060 - 0x0080)
class U__EOSVoiceManager_TA__HandleRemotePlayerJoinedVoiceRoom_0x1 : public UObject
{
public:
class FString          EpicAccountId;    // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString          RoomName;        // 0x0070 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__HandleRemotePlayerJoinedVoiceRoom_0x1");
}

return uClassPointer;
};

void __EOSVoiceManager_TA__HandleRemotePlayerJoinedVoiceRoom_0x1(class
UOnlinePlayer_X* Op);
};
```

```

// Class TAGame.__EOSVoiceManager_TA__HandleRequestedLinkedAccounts_0x1
// 0x0020 (0x0060 - 0x0080)
class U__EOSVoiceManager_TA__HandleRequestedLinkedAccounts_0x1 : public UObject
{
public:
    class FString EpicAccountId; // 0x0060 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)
    class FString RoomName; // 0x0070 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceManager_TA__HandleRequestedLinkedAccounts_0x1");
        }

        return uClassPointer;
    };

    void __EOSVoiceManager_TA__HandleRequestedLinkedAccounts_0x2(class UOnlinePlayer_X*
Op);
    bool __EOSVoiceManager_TA__HandleRequestedLinkedAccounts_0x1(struct
FLinkedAccountData P);
};

// Class TAGame.__EOSVoiceSettingsSave_TA__GetPreferredInputDevice_0x1
// 0x0001 (0x0060 - 0x0061)
class U__EOSVoiceSettingsSave_TA__GetPreferredInputDevice_0x1 : public UObject
{
public:
    uint8_t CurrentPlatform; // 0x0060 (0x0001)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceSettingsSave_TA__GetPreferredInputDevice_0x1");
        }

        return uClassPointer;
    };

    bool __EOSVoiceSettingsSave_TA__GetPreferredInputDevice_0x1(struct FPlatformAudioDevices

```

```

P);
};

// Class TAGame.SaveObject_TA
// 0x0068 (0x0060 - 0x00C8)
class USaveObject_TA : public UObject
{
public:
int32_t Version; // 0x0060 (0x0004)
[0x0000000000000000]
struct FScriptDelegate __EventDirtied__Delegate; // 0x0068 (0x0018)
[0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventLoaded__Delegate; // 0x0080
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventUnloaded__Delegate; // 0x0098
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate EventPropertyChange; // 0x00B0 (0x0018)
[0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SaveObject_TA");
}

return uClassPointer;
};

void EventPropertyChangeFunc();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void UpdateVersion();
class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
bool ShouldForceSave();
void SetDirtyDelayed();
void SetDirty();
void OnUnload();
void Unload();
void OnLoad();
void Load();
void OnCreate();
class USaveObject_TA* Create();
void EventUnloaded(class USaveObject_TA* SaveObject);
void EventLoaded(class USaveObject_TA* SaveObject);
void EventDirtied(class USaveObject_TA* SaveObject);
};

// Class TAGame.EOSVoiceSettingsSave_TA
// 0x0120 (0x00C8 - 0x01E8)
class UEOSVoiceSettingsSave_TA : public USaveObject_TA
{

```

```

public:
unsigned long          bEnabled : 1;           // 0x00C8 (0x0004)
[0x0009000000000000] [0x00000001]
unsigned long          bPushToTalk : 1;        // 0x00C8 (0x0004)
[0x0001000000000000] [0x00000002]
unsigned long          bLocalMuted : 1;        // 0x00C8 (0x0004)
[0x0001000000000000] [0x00000004]
unsigned long          bMainMenuNotifications : 1; // 0x00C8 (0x0004)
[0x0009000000000000] [0x00000008]
unsigned long          bMatchNotifications : 1; // 0x00C8 (0x0004)
[0x0009000000000000] [0x00000010]
unsigned long          bTextNotifications : 1; // 0x00C8 (0x0004)
[0x0001000000000000] [0x00000020]
uint8_t                VoiceChatFilter;       // 0x00CC (0x0001)
[0x0009000000000000]
uint8_t                Voicemode;             // 0x00CD (0x0001)
[0x0009000000000000]
uint8_t                PreferredRoomType;     // 0x00CE (0x0001)
[0x0001000000000000]
float                 OutputVolume;          // 0x00D0 (0x0004)
[0x0009000000000000]
TArray<struct FPlatformAudioDevices> PreferredAudioDevices; // 0x00D8
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)
TArray<class FString> MutedPlayers;         // 0x00E8 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
class FString          InputDeviceId;         // 0x00F8 (0x0010)
[0x000900000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString          OutputDeviceId;        // 0x0108 (0x0010)
[0x000900000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString          Defaultaudioid;        // 0x0118 (0x0010)
[0x000100000400002] (CPF_Const | CPF_NeedCtorLink)
struct FScriptDelegate _bEnabled_ChangeNotify; // 0x0128
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bMainMenuNotifications_ChangeNotify; // 0x0140
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bMatchNotifications_ChangeNotify; // 0x0158
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _VoiceChatFilter_ChangeNotify; // 0x0170
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _Voicemode_ChangeNotify; // 0x0188
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _OutputVolume_ChangeNotify; // 0x01A0
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _InputDeviceId_ChangeNotify; // 0x01B8
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _OutputDeviceId_ChangeNotify; // 0x01D0
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.EOSVoiceSettingsSave_TA");
}

return uClassPointer;
};

void __EOSVoiceSettingsSave_TA__GetVersionDelegates_0x2(class UObject* SaveObj);
void __EOSVoiceSettingsSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __OutputDeviceId_ChangeNotifyFunc();
void __InputDeviceId_ChangeNotifyFunc();
void __OutputVolume_ChangeNotifyFunc();
void __VoiceInputMode_ChangeNotifyFunc();
void __VoiceChatFilter_ChangeNotifyFunc();
void __bMatchNotifications_ChangeNotifyFunc();
void __bMainMenuNotifications_ChangeNotifyFunc();
void __bEnabled_ChangeNotifyFunc();
bool IsVoiceEnabled();
bool IsMuted(class FString EpicAccountId);
void RemoveMutedPlayer(class FString EpicAccountId);
void AddMutedPlayer(class FString EpicAccountId);
uint8_t GetCurrentPlatform();
void SetPreferredOutputDevice(class FString DeviceID);
void SetPreferredInputDevice(class FString DeviceID);
class FString GetPreferredOutputDevice();
class FString GetPreferredInputDevice();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void OnLoad();
void OnCreate();
};

// Class TAGame.__EOSVoiceSettingsSave_TA__GetPreferredOutputDevice_0x1
// 0x0001 (0x0060 - 0x0061)
class U__EOSVoiceSettingsSave_TA__GetPreferredOutputDevice_0x1 : public UObject
{
public:
    uint8_t CurrentPlatform; // 0x0060 (0x0001)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceSettingsSave_TA__GetPreferredOutputDevice_0x1");
        }
    }

    return uClassPointer;
};

bool __EOSVoiceSettingsSave_TA__GetPreferredOutputDevice_0x1(struct

```

```

FPlatformAudioDevices P);
};

// Class TAGame.__EOSVoiceTokenCache_TA__ClearPlayerCachedVoiceCredentials_0x1
// 0x0058 (0x0060 - 0x00B8)
class U__EOSVoiceTokenCache_TA__ClearPlayerCachedVoiceCredentials_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;                      // 0x0060 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString                 RoomId;                      // 0x00A8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceTokenCache_TA__ClearPlayerCachedVoiceCredentials_0x1");
}

return uClassPointer;
};

bool __EOSVoiceTokenCache_TA__ClearPlayerCachedVoiceCredentials_0x1(struct
FPlayerVoiceRoomCredentials P);
};

// Class TAGame.EOSVoiceTokenCache_TA
// 0x0068 (0x0060 - 0x00C8)
class UEOSVoiceTokenCache_TA : public UObject
{
public:
TArray<struct FPlayerVoiceRoomCredentials>   CachedVoiceCredentials;      //
0x0060 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FVoiceRoomTokenRequest>          StagedVoiceCredentials;    //
0x0070 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<class URPC_RequestMatchVoiceTokens_TA*> PendingTokenRequests;     //
0x0080 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class UEOSVoiceConfig_TA*                     VoiceConfig;                // 0x0090 (0x0008)
[0x0001800000000000]
struct FScriptDelegate             __EventVoiceTokenResponse__Delegate; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __EventVoiceTokenRequestError__Delegate; // 0x00B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSVoiceTokenCache_TA");
}

return uClassPointer;
};

void __EOSVoiceTokenCache_TA__HandleVoiceRoomTokens_0x1(struct
FPlayerVoiceRoomCredentials P);
bool TokenRequestIsPending(struct FUniqueNetId PlayerID, class FString RoomId);
void ClearPlayerCachedVoiceCredentials(struct FUniqueNetId PlayerID, class FString RoomId);
void ClearAllCachedVoiceCredentials();
void HandleVoiceRoomTokens(class URPC_RequestMatchVoiceTokens_TA* RPC);
void RequestVoiceRoomTokens();
void RequestVoiceRoomToken(struct FUniqueNetId PlayerID, class FString RoomId);
void EventVoiceTokenRequestError(TArray<struct FVoiceRoomTokenRequest> Requests, class
UError* VoiceError);
void EventVoiceTokenResponse(TArray<struct FVoiceRoomTokenRequest> Requests,
TArray<struct FPlayerVoiceRoomCredentials> Received);
};

// Class TAGame.__EOSVoiceTokenCache_TA__RequestVoiceRoomToken_0x1
// 0x0058 (0x0060 - 0x00B8)
class U__EOSVoiceTokenCache_TA__RequestVoiceRoomToken_0x1 : public UObject
{
public:
class FString RoomId; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FUniqueNetId PlayerID; // 0x0070 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EOSVoiceTokenCache_TA__RequestVoiceRoomToken_0x1");
}

return uClassPointer;
};

bool __EOSVoiceTokenCache_TA__RequestVoiceRoomToken_0x1(struct
FPlayerVoiceRoomCredentials P);
};

// Class TAGame.__EpicAccountSave_TA__GetConvertPlatformFriends_0x1
// 0x0008 (0x0060 - 0x0068)
class U__EpicAccountSave_TA__GetConvertPlatformFriends_0x1 : public UObject
{

```

```
public:
class UOnlineSubsystem* OSS; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EpicAccountSave_TA__GetConvertPlatformFriends_0x1");
}

return uClassPointer;
};

bool __EpicAccountSave_TA__GetConvertPlatformFriends_0x1(struct
FEpicConvertFriendsSaveData Data);
};

// Class TAGame.EpicAccountSave_TA
// 0x0030 (0x00C8 - 0x00F8)
class UEpicAccountSave_TA : public USaveObject_TA
{
public:
TArray<uint8_t> LinkedPlatforms; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FEpicConvertFriendsSaveData> ConvertFriendsData; // 0x00D8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<class FString> SentEOSFriendInvites; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EpicAccountSave_TA");
}

return uClassPointer;
};

void __EpicAccountSave_TA__Reconcile_0x4(class FString FriendId);
void __EpicAccountSave_TA__Reconcile_0x3(uint8_t Platform);
void __EpicAccountSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
void SetConvertPlatformFriends(unsigned long bValue);
bool GetHasPromptedForFriendConversion();
```

```
bool GetConvertPlatformFriends();
};

// Class TAGame.__EpicAccountSave_TA__GetHasPromptedForFriendConversion_0x1
// 0x0008 (0x0060 - 0x0068)
class U_EpicAccountSave_TA__GetHasPromptedForFriendConversion_0x1 : public UObject
{
public:
class UOnlineSubsystem* OSS; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EpicAccountSave_TA__GetHasPromptedForFriendConversion_0x1");
}

return uClassPointer;
};

bool __EpicAccountSave_TA__GetHasPromptedForFriendConversion_0x1(struct
FEpicConvertFriendsSaveData Data);
};

// Class TAGame.__EpicAccountSave_TA__Reconcile_0x1
// 0x0001 (0x0060 - 0x0061)
class U_EpicAccountSave_TA__Reconcile_0x1 : public UObject
{
public:
uint8_t Platform; // 0x0060 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__EpicAccountSave_TA__Reconcile_0x1");
}

return uClassPointer;
};

bool __EpicAccountSave_TA__Reconcile_0x2(struct FEpicConvertFriendsSaveData Data);
bool __EpicAccountSave_TA__Reconcile_0x1(struct FEpicConvertFriendsSaveData Data);
};
```

```
// Class TAGame.__EpicAccountSave_TA_SetConvertPlatformFriends_0x1
// 0x0008 (0x0060 - 0x0068)
class U__EpicAccountSave_TA_SetConvertPlatformFriends_0x1 : public UObject
{
public:
class UOnlineSubsystem* OSS; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EpicAccountSave_TA_SetConvertPlatformFriends_0x1");
}

return uClassPointer;
};

bool __EpicAccountSave_TA_SetConvertPlatformFriends_0x2(struct
FEpicConvertFriendsSaveData Data);
bool __EpicAccountSave_TA_SetConvertPlatformFriends_0x1(struct
FEpicConvertFriendsSaveData Data);
};

// Class TAGame.__EpicFriendsConverterSystem_TA_SetupEpicFriendsConverter_0x1
// 0x0008 (0x0060 - 0x0068)
class U__EpicFriendsConverterSystem_TA_SetupEpicFriendsConverter_0x1 : public UObject
{
public:
class UEpicFriendsConverter_TA* Converter; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__EpicFriendsConverterSystem_TA_SetupEpicFriendsConverter_0x1");
}

return uClassPointer;
};

void __EpicFriendsConverterSystem_TA_SetupEpicFriendsConverter_0x1(class
UEOS_GetAccountsResponse* Response, class UError* Error);
};
```

```
// Class TAGame.EpicFriendsConverter_TA
// 0x0048 (0x0060 - 0x00A8)
class UEpicFriendsConverter_TA : public UObject
{
public:
unsigned long          bCheckedToConvert : 1;           // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]
class UEpicAccountSave_TA*      EpicSave;           // 0x0068 (0x0008)
[0x0000000000000000]
TArray<struct FUniqueNetId>    InvitesToSend;       // 0x0070 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t                  LocalPlayerId;        // 0x0080 (0x0004)
[0x0000000000000000]
class UEpicFriendsConverterConfig_TA*   EpicFriendsConverterConfig; // 
0x0088 (0x0008) [0x0000800000000000]
class UOnlineSubsystem*        EOSSubsystem;        // 0x0090 (0x0008)
[0x0000000000000000]
class UEpicFriendsPlugin_X*    FriendsPlugin;       // 0x0098 (0x0008)
[0x0000000000000000]
float                      SecondsBetweenInvites; // 0x00A0 (0x0004)
[0x0000000000000000]
int32_t                  NumFailedAttempts; // 0x00A4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EpicFriendsConverter_TA");
}

return uClassPointer;
};

void StopConversion();
void RetryOrAbort(class FString& FriendId);
void HandleOnFriendAdded(class UEOS_ManageFriendsListResponse* Response, class UError* Error, class FString& FriendId);
void SendNextFriendRequest();
void SetConversionTimer(float TimeBetweenInvites);
void StartConversion();
void HandleOnReceivedLinkedAccount(unsigned long bSuccess, TArray<struct FLinkedAccountData> LinkedAccountData);
void AddPlatformFriendsFromLinkedAccountMap(class ULinkedAccountMap_X* PlatformFriendsToEpicAccountMap);
void AddPlatformFriendsFromPersonaList(TArray<struct FPersonaDataId> PlatformFriends);
void SetConvertPlatformFriends(unsigned long bConvertPlatformFriends);
bool ShouldPromptToConvertFriends();
};

// Class TAGame.EpicFriendsConverterSystem_TA
```

```

// 0x0000 (0x0060 - 0x0060)
class UEpicFriendsConverterSystem_TA : public UObject
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EpicFriendsConverterSystem_TA");
}

return uClassPointer;
};

static void CheckToConvertPlatformFriends(class UEpicFriendsConverter_TA* InConverter, class ULinkedAccountMap_X* PlatformFriendsToEpicAccountMap);
static void RemoveOutgoingInvites(class UEpicFriendsConverter_TA* Converter, class UEOS_GetAccountsResponse* Response, class UError* Error);
static void SetupEpicFriendsConverter(class UOnlinePlayerFriends_TA* OnlinePlayerFriends, class UGFxShell_TA* InShell, class UOnlineSubsystem* InEOS, class UEpicFriendsPlugin_X* InFriendsPlugin, class UEpicFriends_X* InEpicFriends, class UEpicAccountSave_TA* InSave, class UEOSPermissions_TA* Permissions);
};

// Class TAGame.__ESportConfig_TA__DownloadEventImage_0x1
// 0x0010 (0x0060 - 0x0070)
class U_ESportConfig_TA__DownloadEventImage_0x1 : public UObject
{
public:
class FString URL; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ESportConfig_TA__DownloadEventImage_0x1");
}

return uClassPointer;
};

void __ESportConfig_TA__DownloadEventImage_0x1(class UTexture2DDynamic* Texture);
};

// Class TAGame.ESportConfig_TA

```

```
// 0x0010 (0x0078 - 0x0088)
class UESportConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FESportEventData>          Events;           // 0x0078 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ESportConfig_TA");
}

return uClassPointer;
};

static uint64_t GetNextEventTime(uint64_t CurrentTime, TArray<struct FESportEventData>& InEvents);
static TArray<uint64_t> EventToEpochTimes(struct FESportEventData Event);
void UpdateEvents();
void HandleImageDownloaded(class UTexture2DDynamic* Texture, class FString URL);
void DownloadEventImage(class FString URL);
void Apply();
void eventConstruct();
};

// Class TAGame.__ESportConfig_TA__GetNextEventTime_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ESportConfig_TA__GetNextEventTime_0x1 : public UObject
{
public:
uint64_t          CurrentTime;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ESportConfig_TA__GetNextEventTime_0x1");
}

return uClassPointer;
};

bool __ESportConfig_TA__GetNextEventTime_0x1(uint64_t Time);
};
```

```
// Class TAGame.__Eula_TA__RequiresAcceptance_0x1
// 0x0001 (0x0060 - 0x0061)
class U_Eula_TA__RequiresAcceptance_0x1 : public UObject
{
public:
    uint8_t PlayerPlatform; // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__Eula_TA__RequiresAcceptance_0x1");
        }

        return uClassPointer;
    };

    bool __Eula_TA__RequiresAcceptance_0x1(uint8_t P);
};

// Class TAGame.LegalText_TA
// 0x0030 (0x0060 - 0x0090)
class ULegalText_TA : public UObject
{
public:
    int32_t LegalHash; // 0x0060 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    class FString LegalText; // 0x0068 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    struct FScriptDelegate __EventLegalTextUpdated__Delegate; // 0x0078
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.LegalText_TA");
        }

        return uClassPointer;
    };

    void Accept();
    bool RequiresAcceptance();
    class FString GetText();
    void InitLegal(class FString InLegalText);
```

```

class UUISavedValues_TA* GetUISave();
void EventLegalTextUpdated();
};

// Class TAGame.Eula_TA
// 0x0008 (0x0090 - 0x0098)
class UEula_TA : public ULegalText_TA
{
public:
class UEulaConfig_TA*           EulaConfig;          // 0x0090 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Eula_TA");
}

return uClassPointer;
};

void __Eula_TA__Construct_0x2(class ULocalPlayer_X* LP);
void __Eula_TA__Construct_0x1(class UOnlineGame_X* OnlineGame);
void ResetAcceptance(int32_t Counter);
void Accept();
bool RequiresAcceptance();
class UEulaSave_TA* GetEulaSave();
void HandleNewLegalText(class UOnlineLegalText_X* OnlineLegalText);
void HandleEulaSaveLoaded(class UEulaSave_TA* EulaSave);
void eventConstruct();
};

// Class TAGame.FirstTimeExperienceManager_TA
// 0x0108 (0x0060 - 0x0168)
class UFirstTimeExperienceManager_TA : public UObject
{
public:
uint8_t              Versioning;          // 0x0060 (0x0001)
[0x0000000000000000]
TArray<struct FFTEGroup>      FTEGroups;          // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FFTELevelData>    FTELevelInfo;        // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UFirstTimeExperienceRedefinition_TA*   FTERefine;          // 0x0088
(0x0008) [0x0000800000000000]
struct FFTEGroup            CurrentActiveGroup;    // 0x0090 (0x0048)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                  CurrentCheckpointIndex; // 0x00D8 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FName                CurrentCheckpoint;     // 0x00DC (0x0008)

```

```

[0x0000000000000000] (CPF_Transient)
class UFirstTimeExperienceSave_TA*          FTESave;           // 0x00E8
(0x0008) [0x0000000000000000]
class UPlayerLegacyStatusSave_TA*          LegacySave;        // 0x00F0
(0x0008) [0x0000000000000000]
TArray<struct FFTEChallengeInfo>          ChallengeCompleteInfo; // 0x00F8
(0x0010) [0x0000000004000000] (CPF_NeedCtorLink)
TArray<class UTriggerCondition_TA*>       ActiveTriggers;    // 0x0108
(0x0010) [0x0000000004000000] (CPF_NeedCtorLink)
unsigned long                  bEnabled : 1;           // 0x0118 (0x0004)
[0x0000004000000000] [0x00000001]
struct FScriptDelegate      __EventFTEChanged__Delegate; // 0x0120
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventFTECheckpointCompleted__Delegate; // 0x0138 (0x0018) [0x0000000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventFTEGroupCompleted__Delegate; // 0x0150
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FirstTimeExperienceManager_TA");
}

return uClassPointer;
};

bool __FirstTimeExperienceManager_TA__SetupTriggers_0x2(struct FFTEGroup H);
bool __FirstTimeExperienceManager_TA__SetupTriggers_0x1(struct FFTEGroup G);
void __FirstTimeExperienceManager_TA__ClearTriggers_0x1(class UTriggerCondition_TA* T);
bool __FirstTimeExperienceManager_TA__VerifyVersioning_0x1(struct FCompletedFTEInfo F);
bool __FirstTimeExperienceManager_TA__UpdateToCurrentVersion_0x1(struct FCompletedFTEInfo P);
void DebugPrintLevelInfo();
void DebugPrint();
void DebugPrintGroups();
class UFirstTimeExperienceMetrics_TA* GetMetrics();
void DebugCompleteActiveCheckpoint(unsigned long blsTentativeCompletion);
void DebugCompleteCheckpoint(struct FName GroupName, struct FName CheckpointName);
void DebugSetCheckpoint(struct FName GroupName, struct FName CheckpointName);
struct FFTELevelData GetFTELevelData(class FString InLevelId);
bool HasOtherFTEGroupBlockers(struct FName GroupName);
void NotifyWhenGroupComplete(struct FName GroupName, struct FScriptDelegate Callback);
bool IsActive();
void ClearGroup();
int32_t GetGroupVersion(struct FName GroupName);
void CompleteGroup(struct FName GroupName, int32_t GroupVersion);
void CompleteActiveGroup();
void CompleteActiveCheckpoint(struct FName InCheckpoint);
void CompleteCheckpoint(struct FName GroupName, struct FName CheckpointName, unsigned

```

```

long bAutoCompleted);
void HandlePsyNetLogin();
void SendChallengeInfo(struct FName InGroupName, struct FName InCheckpointName);
void TentativeCompleteActiveCheckpoint(struct FName InCheckpoint);
void SetUserType(uint8_t UserType);
bool IsGroupForUserType(int32_t GroupIndex);
bool GetValidatedGroup(struct FName GroupName, struct FFTEGroup& ValidGroup);
bool TryStartGroup(struct FName GroupName, unsigned long bIgnoreCompletion);
void InitFromSave();
void UpdateToCurrentVersion();
void CompleteEntireGroup(struct FFTEGroup Group);
void VerifyVersioning();
void SetupTrigger(struct FFTEGroup Group);
void ClearTriggers();
void SetupTriggers();
void HandleFTESaveAdded(class UFirstTimeExperienceSave_TA* InFTESave);
void OnUserTypeChanged();
bool UseCommandlineOverride();
void HandleLegacySaveAdded(class UPlayerLegacyStatusSave_TA* InLegacySave);
void HandleRedefinitionChange();
void Init(class ULocalPlayer_TA* LP);
void EventFTETriggerCompleted(struct FName GroupName);
void EventFTECheckpointCompleted(struct FName GroupName, struct FName CheckpointName,
int32_t GroupVersion, unsigned long bAutoComplete);
void EventFTEChanged(class UFirstTimeExperienceManager_TA* FTEManager, unsigned long
bIsActive, struct FName GroupName, struct FName CheckpointName);
};


```

```

// Class TAGame.FirstTimeExperienceSave_TA
// 0x0028 (0x00C8 - 0x00F0)
class UFirstTimeExperienceSave_TA : public USaveObject_TA
{
public:
TArray<struct FCompletedFTETInfo> CompletedFTETs; // 0x00C8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
uint8_t Versioning; // 0x00D8 (0x0001)
[0x0000000000000000]
uint8_t LegacyGroupType; // 0x00D9 (0x0001)
[0x0000000000000000]
struct FName ActiveGroup; // 0x00DC (0x0008)
[0x0000000000000000]
struct FName ActiveCheckpoint; // 0x00E4 (0x0008)
[0x0000000000000000]
unsigned long bTentativelyComplete : 1; // 0x00EC (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FirstTimeExperienceSave_TA");

```

```

}

return uClassPointer;
};

void DebugPrint();
int32_t GetGroupVersion(struct FName GroupName);
void SetActiveCheckpoint(struct FName InCheckpoint);
void SetActiveGroup(struct FName GroupName, struct FName FirstCheckpoint, int32_t InVersion);
bool HasCompletedGroup(struct FName GroupName);
bool HasCompletedCheckpoint(struct FName GroupName, struct FName CheckpointName);
void CompleteCheckpoint(struct FName GroupName, struct FName CheckpointName);
void MovingToCheckpoint(struct FName NewCheckpoint);
void ClearActiveFTE();
void CompleteGroup(struct FName GroupName, int32_t InVersion);
int32_t GetGroupIndex(struct FName GroupName);
bool ShouldForceSave();
};

// Class TAGame.FirstTimeExperienceMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UFirstTimeExperienceMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FirstTimeExperienceMetrics_TA");
}

return uClassPointer;
};

void ActiveCheckpointCompletion(struct FName Group, struct FName Checkpoint, unsigned long bGroupCompleted);
void InitFromRedefinition(struct FName Group, struct FName Checkpoint, unsigned long bIsActive);
void InitFromLoadedSave(struct FName Group, struct FName Checkpoint, unsigned long bIsActive);
};

// Class TAGame.__FirstTimeExperienceManager_TA__DebugSetCheckpoint_0x1
// 0x0008 (0x0060 - 0x0068)
class U__FirstTimeExperienceManager_TA__DebugSetCheckpoint_0x1 : public UObject
{
public:
struct FName                                     GroupName;                      // 0x0060 (0x0008)
[0x0000000000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__FirstTimeExperienceManager_TA__DebugSetCheckpoint_0x1");
}

return uClassPointer;
};

bool __FirstTimeExperienceManager_TA__DebugSetCheckpoint_0x1(struct FFTEGroup P);
};

// Class TAGame.StatusTrigger_FTEGroupComplete_TA
// 0x0008 (0x0080 - 0x0088)
class UStatusTrigger_FTEGroupComplete_TA : public UStatusTrigger_X
{
public:
struct FName ConditionalValue; // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_FTEGroupComplete_TA");
}

return uClassPointer;
};

void DebugPrint();
void HandleChange(struct FName CompletedGroup);
};

// Class TAGame.__FirstTimeExperienceManager_TA__NotifyWhenGroupComplete_0x1
// 0x0020 (0x0060 - 0x0080)
class U__FirstTimeExperienceManager_TA__NotifyWhenGroupComplete_0x1 : public UObject
{
public:
struct FScriptDelegate Callback; // 0x0060 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FName GroupName; // 0x0078 (0x0008)
[0x0000000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._FirstTimeExperienceManager_TA__NotifyWhenGroupComplete_0x1");
}

return uClassPointer;
};

void __FirstTimeExperienceManager_TA__NotifyWhenGroupComplete_0x1(class
UTriggerCondition_TA* _);
};

// Class TAGame.RPC_Challenge_FTEGroupComplete_TA
// 0x0050 (0x00E8 - 0x0138)
class URPC_Challenge_FTEGroupComplete_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FName                   GroupName;         // 0x0130 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_FTEGroupComplete_TA");
}

return uClassPointer;
};

class URPC_Challenge_FTEGroupComplete_TA* SetGroupName(struct FName InGroupName);
class URPC_Challenge_FTEGroupComplete_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_Challenge_FTECheckpointComplete_TA
// 0x0058 (0x00E8 - 0x0140)
class URPC_Challenge_FTECheckpointComplete_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FName                   GroupName;         // 0x0130 (0x0008)
[0x0001000000000000]
struct FName                   CheckpointName;    // 0x0138 (0x0008)
};

```

```
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_Challenge_FTECheckpointComplete_TA");
}

return uClassPointer;
};

class URPC_Challenge_FTECheckpointComplete_TA* SetCheckpointName(struct FName
InCheckpointName);
class URPC_Challenge_FTECheckpointComplete_TA* SetGroupName(struct FName
InGroupName);
class URPC_Challenge_FTECheckpointComplete_TA* SetPlayerID(struct FUniqueNetId
InPlayerId);
};

// Class TAGame.__FirstTimeExperienceManager_TA__SetupTrigger_0x1
// 0x0050 (0x0060 - 0x00B0)
class U__FirstTimeExperienceManager_TA__SetupTrigger_0x1 : public UObject
{
public:
class UTriggerCondition_TA* Condition; // 0x0060 (0x0008)
[0x0000000000000000]
struct FFTEGroup Group; // 0x0068 (0x0048)
[0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__FirstTimeExperienceManager_TA__SetupTrigger_0x1");
}

return uClassPointer;
};

void __FirstTimeExperienceManager_TA__SetupTrigger_0x2(class UTriggerCondition_TA* _);
void __FirstTimeExperienceManager_TA__SetupTrigger_0x1(class UTriggerClump_TA* C);
};

// Class TAGame.FirstTimeExperienceRedefinition_TA
// 0x0014 (0x0060 - 0x0074)
```

```

class UFirstTimeExperienceRedefinition_TA : public UObject
{
public:
TArray<struct FRedefinedInfo>          FTERedefinedInfo;           // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long                  bFTEManagerEnabled : 1;           // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long                  bClearActiveFTE : 1;           // 0x0070 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FirstTimeExperienceRedefinition_TA");
}

return uClassPointer;
};

void DebugPrint();
void RedefineFTESave(class UFirstTimeExperienceSave_TA*& OutSave);
void RedefineFTEGroups(TArray<struct FFTEGroup>& FTEGroups);
};

// Class TAGame.PlayerLegacyStatusSave_TA
// 0x0030 (0x00C8 - 0x00F8)
class UPlayerLegacyStatusSave_TA : public USaveObject_TA
{
public:
uint8_t                      LegacyStatus;           // 0x00C8 (0x0001)
[0x0000000000000000]
unsigned long                 bOnlineSet : 1;           // 0x00CC (0x0004)
[0x0008000000000000] [0x00000001]
TArray<struct FOnlineProductData>      OnlineProducts;        // 0x00D0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __bOnlineSet__ChangeNotify; // 0x00E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerLegacyStatusSave_TA");
}

return uClassPointer;
};

```

```

void __bOnlineSet_ChangeNotifyFunc();
};

// Class TAGame.OnlinePlayer_TA
// 0x0148 (0x0290 - 0x03D8)
class UOnlinePlayer_TA : public UOnlinePlayer_X
{
public:
    class UAchievementManager_TA*           AchievementManagerArchetype;      // 0x0290 (0x0008) [0x0000000000000000]
    class UAchievementManager_TA*           AchievementManager;                // 0x0298 (0x0008) [0x0000000000000000]
    class UNotificationManager_TA*         NotificationManager;             // 0x02A0 (0x0008) [0x0000000000000000]
    class UFirstTimeExperienceManager_TA*   FirstTimeManagerArchetype;        // 0x02A8 (0x0008) [0x0000000000000000]
    class UFirstTimeExperienceManager_TA*   FTEManger;                      // 0x02B0 (0x0008) [0x0000000000000000]
    class UStatusTriggerManager_TA*         TriggerManager;                 // 0x02B8 (0x0008) [0x0000000000000000]
    class UEOSMetrics_X*                  EOSMetricHeartbeat;              // 0x02C0 (0x0008) [0x0001000000000000]
    class UCrumbTrails_TA*                CrumbTrailArchetype;            // 0x02C8 (0x0008) [0x0001000000000000]
    class UCrumbTrails_TA*                CrumbsMgr;                     // 0x02D0 (0x0008) [0x0001000000000000]
    class UChallengeManager_TA*           ChallengeManager;               // 0x02D8 (0x0008) [0x0001000000000000]
    class UWallet_TA*                    WalletArchetype;                // 0x02E0 (0x0008) [0x0000000000000000]
    class UWallet_TA*                    Wallet;                         // 0x02E8 (0x0008) [0x0000000000000000]
    class UShopsManager_TA*              ShopsManagerArchetype;          // 0x02F0 (0x0008) [0x0001000000000000]
    class UShopsManager_TA*              ShopsManager;                   // 0x02F8 (0x0008) [0x0001000000000000]
    class UChatHistory_TA*              ChatHistory;                   // 0x0300 (0x0008) [0x0000000000000000]
    class UOnlinePlayerMTX_TA*          MTX;                           // 0x0308 (0x0008) [0x0000000000000000]
    class UObjectProvider*              RewardDrops;                   // 0x0310 (0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UAccountSettingsComponent_TA* AccountSettings;             // 0x0318 (0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
    class UAgeGate_TA*                AgeGate;                      // 0x0320 (0x0008) [0x0000000000000000]
    struct FUniqueNetId                StayAsPartyLeaderID;           // 0x0328 (0x0048) [0x00000000402000] (CPF_Transient | CPF_NeedCtorLink)
    class UEOSPermissions_TA*          EOSPermissions;                // 0x0370 (0x0008) [0x0000008000000000]
    unsigned long                      bInCabinMode : 1;             // 0x0378 (0x0004) [0x0000004000000000] [0x00000001]
    unsigned long                      bAskAge : 1;                  // 0x0378 (0x0004) [0x0000004000000000] [0x00000001]

```

```
[0x0000004000000000] [0x00000002]
unsigned long          bAskParentEmail : 1;           // 0x0378 (0x0004)
[0x0000004000000000] [0x00000004]
unsigned long          bConfirmDisplayName : 1;       // 0x0378 (0x0004)
[0x0000004000000000] [0x00000008]
unsigned long          bUnsupportedCorrectiveActionNeeded : 1; // 0x0378
(0x0004) [0x0000004000000000] [0x00000010]
unsigned long          bShowedAgeGate : 1;           // 0x0378 (0x0004)
[0x0000004000000000] [0x00000020]
class FString          Country;                  // 0x0380 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
class UEpicConfig_X*   EpicConfig;               // 0x0390 (0x0008)
[0x0000800000000000]
class UESportConfig_TA*   ESportConfig;            // 0x0398 (0x0008)
[0x0000800000000000]
class FString          InvalidDate;              // 0x03A0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString          InvalidEmail;              // 0x03B0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate   __EventCabinModeSet__Delegate; // 0x03C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlinePlayer_TA");
}

return uClassPointer;
};

void __OnlinePlayer_TA__InitPlayer_0x1(class FString MapName);
void __PsyNetPresence__ChangeNotifyFunc();
static class UError* ConvertError(class UEOS_ErrorResponse* ErrorResponse);
void ShowCabinedReminder();
void ShowXboxGuestIsPrimaryPlayerWarningScreen();
void ForceCloseAgeGate();
void UnsupportedCorrectiveActionVerify();
void EpicDisplayNameScreenDone();
void ShowParentEmailScreen();
void ShowDOBScreen();
void ShowWelcomeToCabinedModeScreen();
void ShowWelcomeNotCabinedModeScreen();
void ShowEpicDisplayNameScreen();
void HandleConfirmDisplayName(class UObject* ResponseAsObject, class UError* Error);
void ConfirmDisplayName();
void HandleRetrieveDisplayName(class UDisplayNameResponse* ResponseAsObject, class UError* Error);
void RetrieveDisplayName();
void SetParentEmail(class FString Email);
```

```
void HandleSetParentEmail(class UEmptyResponse* ResponseAsObject, class UError* Error);
bool ValidateEmail(class FString Email);
void CheckForCorrectiveAction();
void ShowEpicAccountLinkScreen();
void SetAskConfirmDisplayName(unsigned long val);
void SetAskParentEmail(unsigned long val);
bool UnsupportedCorrectiveActionNeeded();
bool AskConfirmDisplayName();
bool AskParentEmail();
bool ShowedAgeGate();
void SetShowedAgeGate(unsigned long val);
bool AskAge();
void SetAskAge(unsigned long val);
void SetInCabinMode(unsigned long bnewValue);
bool IsInCabinMode();
bool EnforceCabinMode();
bool PromptForPin();
bool AreShopNotificationsEnabledDefault();
bool AreShopNotificationsEnabled();
void SetRequirePinForFriends(unsigned long bVal);
bool IsPinRequiredForFriends();
uint8_t GetVoiceChatPermissionDefaultLevel();
uint8_t GetTextChatPermissionDefaultLevel();
uint8_t GetVoiceChatPermissionLevel();
uint8_t GetTextChatPermissionLevel();
bool IsTutorialActive();
void GetPlayersLikesDislikes(class UGameSettingPlaylist_X* Playlist, TArray<struct FName>& PlayerLikes, TArray<struct FName>& PlayerDislikes);
void HandleReceivedController(class UPlayer* PlayerRef);
void OnLoginChange(uint8_t LocalUserNum);
void HandleCountry(class FString InCountry);
void InitPlayer();
void eventConstruct();
void EventCabinModeSet();
};

// Class TAGame.__FirstTimeExperienceSave_TA__GetGroupIndex_0x1
// 0x0008 (0x0060 - 0x0068)
class U__FirstTimeExperienceSave_TA__GetGroupIndex_0x1 : public UObject
{
public:
    struct FName           GroupName;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__FirstTimeExperienceSave_TA__GetGroupIndex_0x1");
        }
    }
}
```

```

return uClassPointer;
};

bool __FirstTimeExperienceSave_TA__GetGroupIndex_0x1(struct FCompletedFTEInfo P);
};

// Class TAGame.__FreeplayCommands_TA__GetDefendShotBreakout_0x1
// 0x000C (0x0060 - 0x006C)
class U_FreeplayCommands_TA__GetDefendShotBreakout_0x1 : public UObject
{
public:
struct FVector CarLocation; // 0x0060 (0x000C)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__FreeplayCommands_TA__GetDefendShotBreakout_0x1");
}

return uClassPointer;
};

int32_t __FreeplayCommands_TA__GetDefendShotBreakout_0x1(class
ABreakOutActor_Platform_TA* A, class ABreakOutActor_Platform_TA* B);
};

// Class TAGame.FreeplayCommands_TA
// 0x0020 (0x0060 - 0x0080)
class UFreeplayCommands_TA : public UObject
{
public:
class AGameEvent_Soccar_TA* SoccarGame; // 0x0060
(0x0008) [0x0001000000000000]
class UFreeplayCommandsConfig_TA* FreeplayConfig; // 0x0068
(0x0008) [0x0001800000000000]
TArray<struct FFreeplayCommandEvent> ActivatedCommands; // 0x0070
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FreeplayCommands_TA");
}
}

```

```

return uClassPointer;
};

struct FVector GetDefendShotStandard(class ABall_TA* Ball, struct FVector CarLocation, class UGoal_TA* ClosestGoal);
struct FVector GetDefendShotBreakout(class ABall_TA* Ball, struct FVector CarLocation, class AGameEvent_Breakout_TA* BreakoutGame);
struct FVector GetDefendShotHoops(class ABall_TA* Ball, struct FVector CarLocation, class UGoal_Hoops_TA* HoopsGoal, struct FVector BallProximityToRim);
class UError* DefendShot(class ABall_TA* Ball, struct FVector CarLocation, class UGoal_TA* ClosestGoal);
class UError* RedirectPass(class ABall_TA* Ball, class ACar_TA* Car, struct FVector BestGoalLocation);
class UError* PopBallUp(class ABall_TA* Ball);
class UError* LaunchBall(class ABall_TA* Ball, struct FVector LaunchVector);
class UError* BallOnCarInAir(class ABall_TA* Ball, class ACar_TA* Car);
class UError* BallOnCar(class ABall_TA* Ball, class ACar_TA* Car);
class UError* BallInFront(class ABall_TA* Ball, class ACar_TA* Car);
class UError* ConditionalTeleportBallToCar(class ABall_TA* Ball, struct FVector TeleportLocation, struct FVector CarLocation, struct FVector NewBallVelocity);
struct FActivateFreeplayCommandOutParams ActivateFreeplayCommand(class APlayerController* Caller, uint8_t CommandToActivate);
void HandleGameEventCommand(class APlayerController* Caller, class FString CommandString);
void SendBatchedMetrics();
class UFreeplayCommandsMetrics_TA* GetMetrics();
bool CanUseFreeplayCommands(int32_t ControllerId);
void HandleSoccerGameDestroyed(class AGameEvent_TA* InGameEvent);
void InitSoccerGame(class AGameEvent_Soccer_TA* InSoccerGame);
};

// Class TAGame.__FXActor_Boost_TA__DuplicateAttachment_0x1
// 0x0014 (0x0060 - 0x0074)
class U__FXActor_Boost_TA__DuplicateAttachment_0x1 : public UObject
{
public:
TArray<class UActorComponent*> ComponentsToUse; // 0x0060
(0x0010) [0x000000004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
int32_t AttachmentIdx; // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__FXActor_Boost_TA__DuplicateAttachment_0x1");
}
}

```

```

return uClassPointer;
};

void __FXActor_Boost_TA__DuplicateAttachment_0x1(struct FName X);
};

// Class TAGame.__FXActor_SafeZone_Knockout_TA__GetActiveSafeZoneIndex_0x1
// 0x0004 (0x0060 - 0x0064)
class U__FXActor_SafeZone_Knockout_TA__GetActiveSafeZoneIndex_0x1 : public UObject
{
public:
int32_t MatchTimeActive; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__FXActor_SafeZone_Knockout_TA__GetActiveSafeZoneIndex_0x1");
}

return uClassPointer;
};

bool __FXActor_SafeZone_Knockout_TA__GetActiveSafeZoneIndex_0x1(struct FSafeZoneInfo S);
};

// Class TAGame.FXActor_SafeZone_Knockout_TA
// 0x0068 (0x03678 - 0x03DE0)
class AFXActor_SafeZone_Knockout_TA : public AFXActor_TA
{
public:
TArray<struct FSafeZoneInfo> SafeZones; // 0x03678 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FSafeZoneInfo> EliminationSafeZones; // 0x03788
(0x0010) [0x0001000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
TArray<struct FSafeZoneInfo> NonEliminationSafeZones; // 0x03898
(0x0010) [0x0001000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
class AGameEvent_KnockOut_TA* GameEvent; // 0x039A8
(0x0008) [0x000100000002000] (CPF_Transient)
int32_t LastZoneIndex; // 0x03AB0 (0x0004)
[0x000100000002000] (CPF_Transient)
TArray<class UFXActorEvent_X*> StateEvents; // 0x03AB8
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventZoneShrinking__Delegate; // 0x03BC8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_SafeZone_Knockout_TA");
}

return uClassPointer;
};

void __FXActor_SafeZone_Knockout_TA__UpdateFXState_0x1(class UFXActorEvent_X* E);
bool ReachedFinalZone();
int32_t ZoneSort(struct FSafeZoneInfo& A, struct FSafeZoneInfo& B);
int32_t GetActiveSafeZoneIndex(int32_t MatchTimeActive);
void UpdateFXState();
void eventTick(float DeltaTime);
void HandleEliminationModeChanged(class AGameEvent_KnockOut_TA* GameEvent_KO);
void SetGameEvent(class AGameEvent_Soccar_TA* InGameEvent);
void EventZoneShrinking();
};

// Class TAGame.__FXActor_TA__ApplyPaint_0x1
// 0x0010 (0x0060 - 0x0070)
class U__FXActor_TA__ApplyPaint_0x1 : public UObject
{
public:
class UProductAttribute_PaintSettings_TA*      PaintSettings;           // 0x0060
(0x0008) [0x0000000000000000]
class UProductPaint_TA*                         Paint;                  // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__FXActor_TA__ApplyPaint_0x1");
}

return uClassPointer;
};

void __FXActor_TA__ApplyPaint_0x1(struct FFXAttachment FXAttach);

// Class TAGame.__FXActor_TA__ApplyPaintToAttachment_0x1
// 0x0010 (0x0060 - 0x0070)
class U__FXActor_TA__ApplyPaintToAttachment_0x1 : public UObject
{
public:
class UProductAttribute_PaintSettings_TA*      PaintSettings;           // 0x0060
(0x0008) [0x0000000000000000]

```

```
class UProductPaint_TA*           Paint;          // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._FXActor_TA__ApplyPaintToAttachment_0x1");
}

return uClassPointer;
};

void __FXActor_TA__ApplyPaintToAttachment_0x1(class UParticleSystemComponent*
BeamPSC);
};

// Class TAGame.__GameEvent_KnockOut_TA__AddPRI_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_KnockOut_TA__AddPRI_0x1 : public UObject
{
public:
class APRI_TA*                  PRI;           // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GameEvent_KnockOut_TA__AddPRI_0x1");
}

return uClassPointer;
};

bool __GameEvent_KnockOut_TA__AddPRI_0x1(class APRI_TA* P);
};

// Class TAGame.GameEvent_KnockOut_TA
// 0x00A0 (0x0CA0 - 0x0D40)
class AGameEvent_KnockOut_TA : public AGameEvent_Soccar_TA
{
public:
unsigned long                   bFreeForAll : 1;      // 0x0CA0 (0x0004)
[0x0001000000000002] [0x00000001] (CPF_Const)
int32_t                          PlayerLives;        // 0x0CA4 (0x0004)
[0x0001004100000020] (CPF_Net)
```

```

int32_t PlayersEliminated; // 0x0CA8 (0x0004)
[0x0001004000000000]
TArray<class ASpecialPickup_TA*> PickupArchetypes; // 0x0CB0
(0x0010) [0x00010000040001] (CPF_Edit | CPF_NeedCtorLink)
class AFXActor_X* PickupFXActor; // 0x0CC0 (0x0008)
[0x0001000000000001] (CPF_Edit)
float PickupRespawnDelay; // 0x0CC8 (0x0004)
[0x0001000000000001] (CPF_Edit)
class AFXActor_SafeZone_Knockout_TA* SafeZone; // 0x0CD0
(0x0008) [0x0001000000000000]
class AFXActor_SafeZone_Knockout_TA* SafeZoneArchetype; // 0x0CD8
(0x0008) [0x0001000000000000]
class UMessage_TA* ReturnToSafezoneMessage; // 0x0CE0
(0x0008) [0x0001000000000000]
class UMessage_TA* ReturnedToSafezoneMessage; // 0x0CE8
(0x0008) [0x0001000000000000]
class UMessage_TA* SafeZoneShrinkingMessage; // 0x0CF0
(0x0008) [0x0001000000000000]
class UMessage_TA* SafeZoneFinalZoneMessage; // 0x0CF8
(0x0008) [0x0001000000000000]
class UMessage_TA* OutOfWorldMessage; // 0xD00 (0x0008)
[0x0001000000000000]
float PodiumSpawnLocationZ; // 0xD08 (0x0004)
[0x0001000000000020] (CPF_Net)
struct FScriptDelegate __EventPlayerStatEvent__Delegate; // 0xD10
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventEliminationModeChanged__Delegate; // 
0xD28 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_KnockOut_TA");
}

return uClassPointer;
};

void GetPodiumHeight();
static bool __GameEvent_KnockOut_TA__GetPRIsWithLivesLeft_0x1(class APRI_TA* PRI);
bool __GameEvent_KnockOut_TA__GetMVP_0x1(class APRI_KnockOut_TA* P);
bool __GameEvent_KnockOut_TA__GetPodiumHeight_0x1(class APRI_TA* PRI);
void AddAdditionalViewableActors(TArray<class FString>& Actors);
void UpdateBotCount();
bool CanSpectate();
bool SplitScoreboardByTeams();
bool IsMatchWinner(class APRI_TA* PRI);
bool IsSuddenKO();
void HandleActorOutOfWorld(class AOutOfWorldVolume_TA* OutOfWorldVolume, class AActor* Actor);

```

```

void HandleSafeZoneShrinking();
void MessageReturnedToSafezone(class APlayerController_TA* Player);
void MessageReturnToSafezone(int32_t TimeLeft, class APlayerController_TA* Player);
bool IsFreeForAll();
int32_t MVPSort(class APRI_TA* A, class APRI_TA* B);
class APRI_TA* GetMVP(class ATeam_TA* WinningTeam);
int32_t ScoreboardSort(class APRI_TA* A, class APRI_TA* B);
bool HasMatchWinner();
bool HasWinner();
bool IsEliminationMode();
static TArray<class APRI_TA*> GetPRIsWithLivesLeft(TArray<class APRI_TA*>& InPRIs);
int32_t GetTeamLifeCounts();
void HandlePlayerKnockedOut(class ACar_KnockOut_TA* Victim);
void RefreshActiveMVP();
void HandlePawnTypeChanged(class APRI_TA* InPRI);
void HandlePlayerStatEvent(class APRI_TA* InPRI, class UStatEvent_TA* StatEvent, int32_t Count);
void RemovePRI(class APRI_TA* PRI);
void AddPRI(class APRI_TA* PRI);
TArray<class APRI_TA*> GetAllPRIs();
bool PlayerCanRestart(class AController* aPlayer);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct FRotator& out_Rotation);
void InitPickups();
void InitPlayerStarts();
void SelectPlayerTeam(class APRI_TA* PRI);
bool AllTeamsCreated();
void CreateTeams();
void OnEliminationModeChanged();
void eventInitGame(class FString Options);
void OnInit();
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
void EventEliminationModeChanged(class AGameEvent_KnockOut_TA* GameEvent);
void EventPlayerStatEvent(class APRI_TA* InPRI, class UStatEvent_TA* StatEvent, int32_t Count);
};

// Class TAGame.__GameEvent_Soccar_TA__AddPRI_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GameEvent_Soccar_TA__AddPRI_0x1 : public UObject
{
public:
    class APRI_TA*          PRI;           // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__GameEvent_Soccar_TA__AddPRI_0x1");
        }
    }
};

```

```

return uClassPointer;
};

bool __GameEvent_Soccar_TA__AddPRI_0x1(class APRI_TA* P);
};

// Class TAGame.__GameEvent_Soccar_TA__ChallengePlayerToHonorDuel_0x1
// 0x0090 (0x0060 - 0x00F0)
class U__GameEvent_Soccar_TA__ChallengePlayerToHonorDuel_0x1 : public UObject
{
public:
struct FUniqueNetId           Challenger;          // 0x0060 (0x0048)
[0x0000000000040000] (CPF_NeedCtorLink)
struct FUniqueNetId           Defender;           // 0x00A8 (0x0048)
[0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__ChallengePlayerToHonorDuel_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__GameEvent_Soccar_TA__CommitPlayerMatchData_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_Soccar_TA__CommitPlayerMatchData_0x1 : public UObject
{
public:
class APRI_TA*                 PRI;                // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__CommitPlayerMatchData_0x1");
}

return uClassPointer;
};

```

```
bool __GameEvent_Soccar_TA__CommitPlayerMatchData_0x1(struct FReservationData Reservation);
};

// Class TAGame.__GameEvent_Soccar_TA__GetGoalByTeamIndex_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GameEvent_Soccar_TA__GetGoalByTeamIndex_0x1 : public UObject
{
public:
int32_t TeamIndex; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__GetGoalByTeamIndex_0x1");
}

return uClassPointer;
};

bool __GameEvent_Soccar_TA__GetGoalByTeamIndex_0x1(class UGoal_TA* G);
};

// Class TAGame.__GameEvent_Soccar_TA__GetShouldStart_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GameEvent_Soccar_TA__GetShouldStart_0x1 : public UObject
{
public:
TArray<struct FUniqueNetId> PlayersAbleToStart; // 0x0060
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__GetShouldStart_0x1");
}

return uClassPointer;
};

void __GameEvent_Soccar_TA__GetShouldStart_0x1(class APRI_TA* PRI);
};
```

```

// Class TAGame.__GameEvent_Soccar_TA__HandlePlayerSkillUpdated_0x1
// 0x00C8 (0x0060 - 0x0128)
class U__GameEvent_Soccar_TA__HandlePlayerSkillUpdated_0x1 : public UObject
{
public:
    struct FUniqueNetId           PlayerID;          // 0x0060 (0x0048)
    [0x0000000000040000] (CPF_NeedCtorLink)
    struct FUpdatedPlayerSkillRating Rating;          // 0x00A8 (0x0080)
    [0x0000000000040000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__HandlePlayerSkillUpdated_0x1");
        }

        return uClassPointer;
    };

void __GameEvent_Soccar_TA__HandlePlayerSkillUpdated_0x1(class APRI_TA* PRI);
};

// Class TAGame.__GameEvent_Soccar_TA__InitClubMatch_0x2
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_Soccar_TA__InitClubMatch_0x2 : public UObject
{
public:
    class UOnlineClubCache_X*      Cache;           // 0x0060 (0x0008)
    [0x0001000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__InitClubMatch_0x2");
        }

        return uClassPointer;
    };

    uint64_t __GameEvent_Soccar_TA__InitClubMatch_0x4(struct FReservationData P);
    uint64_t __GameEvent_Soccar_TA__InitClubMatch_0x2(struct FReservationData P);
};

```

```

// Class TAGame.__GameEvent_Soccar_TA_ReportFps_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_Soccar_TA_ReportFps_0x1 : public UObject
{
public:
float TotalGameTime; // 0x0060 (0x0004)
[0x0000000000000000]
int32_t TotalFrames; // 0x0064 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GameEvent_Soccar_TA_ReportFps_0x1");
}

return uClassPointer;
};

void __GameEvent_Soccar_TA_ReportFps_0x3(struct FFPSBucketMetrics Bucket);
struct FFPSBucketMetrics __GameEvent_Soccar_TA_ReportFps_0x2(struct FFpsBucketData BucketAverage);
void __GameEvent_Soccar_TA_ReportFps_0x1(struct FFpsBucketData Bucket);
};

// Class TAGame.__GameEvent_Soccar_TA_SetGoalsEnabled_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GameEvent_Soccar_TA_SetGoalsEnabled_0x1 : public UObject
{
public:
unsigned long bEnabled : 1; // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA_SetGoalsEnabled_0x1");
}

return uClassPointer;
};

void __GameEvent_Soccar_TA_SetGoalsEnabled_0x1(class UGoal_TA* Goal);
};

```

```
// Class TAGame.__GameEvent_Soccar_TA__ShouldAutoReadyUp_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__GameEvent_Soccar_TA__ShouldAutoReadyUp_0x1 : public UObject
{
public:
struct FUniqueNetId           PrimaryID;                      // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Soccar_TA__ShouldAutoReadyUp_0x1");
}

return uClassPointer;
};

bool __GameEvent_Soccar_TA__ShouldAutoReadyUp_0x1(class APRI_TA* P);
};

// Class TAGame.__GameEvent_TA__GetGeneratedBotName_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GameEvent_TA__GetGeneratedBotName_0x1 : public UObject
{
public:
class FString                 BotName;                      // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_TA__GetGeneratedBotName_0x1");
}

return uClassPointer;
};

bool __GameEvent_TA__GetGeneratedBotName_0x1(class APRI_TA* PRI);
};

// Class TAGame.__GameEvent_TA__HasMutatorNamed_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_TA__HasMutatorNamed_0x1 : public UObject
{
```

```
public:
struct FName MutatorName; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GameEvent_TA__HasMutatorNamed_0x1");
}

return uClassPointer;
};

bool __GameEvent_TA__HasMutatorNamed_0x1(class UMutator_TA* M);
};

// Class TAGame.Mutator_TA
// 0x0004 (0x0060 - 0x0064)
class UMutator_TA : public UObject
{
public:
unsigned long bRecordMetrics : 1; // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_TA");
}

return uClassPointer;
};

void MutateObject(class UObject* O);
void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame._GameEvent_TA__HasMutatorType_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GameEvent_TA__HasMutatorType_0x1 : public UObject
{
public:
class UClass* MutatorClass; // 0x0060 (0x0008)
[0x0000000000000000]
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GameEvent_TA__HasMutatorType_0x1");
}

return uClassPointer;
};

bool __GameEvent_TA__HasMutatorType_0x1(class UMutator_TA* M);
};

// Class TAGame.__GameEvent_TA__HasPlayerNamed_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GameEvent_TA__HasPlayerNamed_0x1 : public UObject
{
public:
class FString PlayerName; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GameEvent_TA__HasPlayerNamed_0x1");
}

return uClassPointer;
};

bool __GameEvent_TA__HasPlayerNamed_0x1(class APRI_TA* PRI);
};

// Class TAGame.__GameEvent_TA__SetBotSkillByTeam_0x1
// 0x000C (0x0060 - 0x006C)
class U__GameEvent_TA__SetBotSkillByTeam_0x1 : public UObject
{
public:
int32_t Team; // 0x0060 (0x0004)
[0x0000000000000000]
float NewSkill; // 0x0064 (0x0004)
[0x0000000000000000]
float BoostThreshold; // 0x0068 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_TA_SetBotSkillByTeam_0x1");
}

return uClassPointer;
};

void __GameEvent_TA_SetBotSkillByTeam_0x2(class AAIController_TA* AI);
bool __GameEvent_TA_SetBotSkillByTeam_0x1(class AAIController_TA* AI);
};

// Class TAGame.__GameEvent_TrainingEditor_TA_HandlePrimaryGameplaySettingsSave_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_TrainingEditor_TA_HandlePrimaryGameplaySettingsSave_0x1 : public
UObject
{
public:
class UGameplaySettingsSave_TA*           Settings;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_TrainingEditor_TA_HandlePrimaryGameplaySettingsSave_0x1");
}

return uClassPointer;
};

void __GameEvent_TrainingEditor_TA_HandlePrimaryGameplaySettingsSave_0x1();
};

// Class TAGame.GameplaySettingsSave_TA
// 0x0168 (0x00C8 - 0x0230)
class UGameplaySettingsSave_TA : public USaveObject_TA
{
public:
unsigned long           bFilterNonTacticalQuickChat : 1;      // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long           bColorBlind : 1;                  // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long           bForceDefaultColors : 1;        // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long           bEnableHorizontalSplitscreen : 1; // 0x00C8 (0x0004)

```

```
[0x0000000000000000] [0x00000008]
unsigned long          bEnableFreeplayUnlimitedBoost : 1;           // 0x00C8
(0x0004) [0x0008000000000000] [0x00000010]
unsigned long          bUseSteamInput : 1;                         // 0x00C8 (0x0004)
[0x0009000000000000] [0x00000020]
unsigned long          bDisableFreeplayGoals : 1;                // 0x00C8 (0x0004)
[0x0008000000000000] [0x00000040]
unsigned long          bIsTradingEnabled : 1;                  // 0x00C8 (0x0004)
[0x0009000000000000] [0x00000080]
unsigned long          bFirstTimeDefaultVoice : 1;             // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000100]
unsigned long          bFirstTimeDefaultText : 1;              // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000200]
unsigned long          bFirstTimeDefaultTrade : 1;            // 0x00C8 (0x0004)
[0x0001000000000000] [0x00000400]
uint8_t                QuickChatFilter;                      // 0x00CC (0x0001)
[0x0008000000000000]
uint8_t                MatchChatFilter;                     // 0x00CD (0x0001)
[0x0008000000000000]
uint8_t                PartyChatFilter;                    // 0x00CE (0x0001)
[0x0008000000000000]
uint8_t                DeprecatedVoiceFilter;        // 0x00CF (0x0001)
[0x0008000000000000]
uint8_t                ChatFilter;                        // 0x00D0 (0x0001)
[0x0008000000000000]
uint8_t                EffectIntensity;                 // 0x00D1 (0x0001)
[0x0000000000000000]
uint8_t                StatEventDisplayLevel;       // 0x00D2 (0x0001)
[0x0000000000004000] (CPF_Config)
uint8_t                FreeplayBoostFillType;        // 0x00D3 (0x0001)
[0x0008000000000000]
uint8_t                StatGraphLevel;                   // 0x00D4 (0x0001)
[0x0000000000000000]
uint8_t                TrainingControlsVisibility; // 0x00D5 (0x0001)
[0x0008000000000000]
float                 SafeZoneRatio;                   // 0x00D8 (0x0004)
[0x0000000000000000]
float                 UIScale;                        // 0x00DC (0x0004)
[0x0000000000000000]
float                 NameplateScale;                 // 0x00E0 (0x0004)
[0x0000000000000000]
float                 TrainingGameSpeed;               // 0x00E4 (0x0004)
[0x0008000000000000]
float                 FreeplayBoostFillDelay;     // 0x00E8 (0x0004)
[0x0008000000000000]
class FString          PreviousDefaultVoice;      // 0x00F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString          PreviousDefaultText;       // 0x0100 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bEnableFreeplayUnlimitedBoost__ChangeNotify; //
0x0110 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bUseSteamInput__ChangeNotify;        // 0x0128
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bDisableFreeplayGoals__ChangeNotify; // 0x0140
```

```

(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __QuickChatFilter_ChangeNotify;      // 0x0158
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __MatchChatFilter_ChangeNotify;    // 0x0170
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __PartyChatFilter_ChangeNotify;   // 0x0188
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __DeprecatedVoiceFilter_ChangeNotify; // 0x01A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __ChatFilter_ChangeNotify;        // 0x01B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __FreeplayBoostFillType_ChangeNotify; // 0x01D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __TrainingControlsVisibility_ChangeNotify; // 0x01E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __TrainingGameSpeed_ChangeNotify;   // 0x0200
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __FreeplayBoostFillDelay_ChangeNotify; // 0x0218
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameplaySettingsSave_TA");
}

return uClassPointer;
};

void __GameplaySettingsSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __FreeplayBoostFillDelay_ChangeNotifyFunc();
void __TrainingGameSpeed_ChangeNotifyFunc();
void __TrainingControlsVisibility_ChangeNotifyFunc();
void __FreeplayBoostFillType_ChangeNotifyFunc();
void __ChatFilter_ChangeNotifyFunc();
void __DeprecatedVoiceFilter_ChangeNotifyFunc();
void __PartyChatFilter_ChangeNotifyFunc();
void __MatchChatFilter_ChangeNotifyFunc();
void __QuickChatFilter_ChangeNotifyFunc();
void __bIsTradingEnabled_ChangeNotifyFunc();
void __bDisableFreeplayGoals_ChangeNotifyFunc();
void __bUseSteamInput_ChangeNotifyFunc();
void __bEnableFreeplayUnlimitedBoost_ChangeNotifyFunc();
void SetNameplateScale(float Value);
void SetUIScale(float Value);
void SetSafeZone(float Value);
void OnLoad();
static uint8_t ConvertMatchFilterToPartyFilter(uint8_t InMatchChatFilter);
static uint8_t ConvertVoiceFilterToChatFilter(uint8_t InVoiceFilter);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);

```

```

void OnCreate();
};

// Class TAGame.GameEvent_GameEditor_TA
// 0x0070 (0x0CA0 - 0x0D10)
class AGameEvent_GameEditor_TA : public AGameEvent_Soccar_TA
{
public:
class UClass* SaveDataClass; // 0x0CA0 (0x0008)
[0x0000000000000000]
class USaveData_GameEditor_TA* SaveData; // 0x0CA8
(0x0008) [0x0000000000000000]
int32_t ActiveRoundNumber; // 0xCB0 (0x0004)
[0x0000000000000000]
TArray<struct FSpawnArchetypeData> SpawnableArchetypes; // 0xCB8
(0x0010) [0x00000000400000] (CPF_NeedCtorLink)
TArray<class ADynamicSpawnPointMesh_TA*> CarSpawnPoints; // 0x0CC8 (0x0010) [0x00000000400000] (CPF_NeedCtorLink)
class AGameEditor_Pawn_TA* EditorPawnArchetype; // 0xCD8
(0x0008) [0x0000000000000000]
int32_t MaxRounds; // 0xCE0 (0x0004)
[0x0000000000000000]
TArray<struct FActorHistory> UndoHistory; // 0xCE8 (0x0010)
[0x00000000400000] (CPF_NeedCtorLink)
int32_t HistoryPosition; // 0xCF8 (0x0004)
[0x0000000000000000]
int32_t MaxUndoHistory; // 0xCFC (0x0004)
[0x0000000000000000]
class AFXActor_X* FXActorArchetype; // 0xD00 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X* FXActor; // 0xD08 (0x0008)
[0x000008000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_GameEditor_TA");
}

return uClassPointer;
};

void HandleTriggerTouched(class AGameEditor_Trigger_TA* Trigger, class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector HitLocation, struct FVector HitNormal);
void RotateActor(class APlayerController_TA* PC, unsigned long bSnapOrientation);
void OnPlayerAdded(class AController* NewPlayer);
void PrevRound();
void NextRound();
void DecreaseTime();
void IncreaseTime();

```

```
void StopEditing(class APlayerController_TA* PC);
void StartEditing();
void CycleActor(class APlayerController_TA* PC);
void ReleaseGrabbedActor(class APlayerController_TA* PC);
void ReleaseRotateActor(class APlayerController_TA* PC);
void ToggleRotateActor(class APlayerController_TA* PC);
void SetSaveData(class USaveData_GameEditor_TA* Data);
void ToggleGrabActor(class APlayerController_TA* PC);
void ToggleReleaseActor(class APlayerController_TA* PC);
void ReleaseActor(class APlayerController_TA* PC);
void GrabActor(class APlayerController_TA* PC);
bool CanQueSaveReplay();
bool ShouldUpdateCrosshair();
int32_t GetPlayerTeamNumber();
bool CanAddHistory();
void ToggleEditorRound();
void CommitRedoRound();
void ResetRound(unsigned long bForceReset);
void Save();
void Redo(class APlayerController_TA* PC);
void Undo(class APlayerController_TA* PC);
int32_t GetRoundNumber();
class AActor* FindActorByName(struct FName actorName);
bool RestoreFromHistory(class APlayerController_TA* PC, uint8_t HistoryType, struct FActorHistory RestoreHistory);
uint8_t GetOtherHistoryType(uint8_t HistoryType);
void ClampUndoHistory();
bool AddToHistoryArray(class UInterface_GameEditor_TA* EditorActor, struct FActorHistory NewHistory);
bool HistoriesAreEqual(class UInterface_GameEditor_TA* EditorActor, struct FActorHistory H0, struct FActorHistory H1);
bool AddUndoHistory(class UInterface_GameEditor_TA* EditorActor, uint8_t HistoryEditType, unsigned long bClearRedoHistory);
bool AddHistory(uint8_t HistoryType, class UInterface_GameEditor_TA* EditorActor, uint8_t HistoryEditType, unsigned long bClearRedoHistory);
struct FActorHistory MakeHistory(class UInterface_GameEditor_TA* EditorActor, uint8_t HistoryEditType);
void ClearRedoHistory();
void ClearAllHistory();
class AActor* DeserializeAndSpawn(class FString SerializedData, class UClass* ClassType, uint8_t HistoryType);
bool DestroyActor(class AActor* A, uint8_t HistoryType);
bool IsInEditMode();
void RemoveAllPointsFromScore(int32_t TeamIndex);
void RemovePointsFromScore(int32_t PointsToRemove, int32_t TeamIndex);
struct FName GetDesiredSoundState();
void DeleteAllExistingActorsBasedOffSpawnList();
int32_t GetSpawnedActorCount(struct FSpawnArchetypeData ArchetypeData);
bool RoundContainsASwitch();
void EnableTriggers(unsigned long bEnable);
void HideCarSpawnPoints(unsigned long bHide);
void ResetSpawnLocations();
void OnSpawnedArchetype(class AActor* SpawnedActor, uint8_t HistoryType);
void SpawnArchetype(class APlayerController_TA* Controller, int32_t ArchetypeIndex);
```

```

struct FSpawnArchetypeData GetSelectedSpawnArchetype(int32_t Index);
void IncrementSelectedSpawnArchetypeIndex(int32_t Direction, int32_t& Index);
void ResetBallsToDefaultPosition();
void FireBalls();
void Interact();
bool ShouldResetBalls();
void OnVehicleSetup(class ACar_TA* Car);
void HandleVehicleSetup(class ACar_TA* Car);
void OnPlayerRestarted(class ACar_TA* PlayerCar);
void GetSpawnLocationAndRotation(class AController* Player, struct FVector& SpawnLocation,
struct FRotator& SpawnRotation);
bool RestartPlayer(class AController* NewPlayer);
class AGameEditor_Pawn_TA* SpawnEditorPawn(class AController* NewPlayer, struct FVector
SpawnLoc, struct FRotator SpawnRot);
void SetupInitialPawnOrientation(class APlayerController* NewPlayer);
bool ChooseTeam(int32_t TeamIndex, class APlayerController_TA* Player);
void InitFX();
void eventPostBeginPlay();
};


```

```

// Class TAGame.GameEvent_TrainingEditor_TA
// 0x0160 (0x0D10 - 0x0E70)
class AGameEvent_TrainingEditor_TA : public AGameEvent_GameEditor_TA
{
public:
float MinRoundTime; // 0x0D10 (0x0004)
[0x0000000000000000] float MaxRoundTime; // 0x0D14 (0x0004)
[0x0000000000000000] unsigned long bNoEditor : 1; // 0x0D18 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient) unsigned long bDisplayedRedoPenaltyMessage : 1; // 0x0D18
(0x0004) [0x0000000000002000] [0x00000002] (CPF_Transient) unsigned long bUnsavedChanges : 1; // 0x0D18 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient) unsigned long bShouldEndTraining : 1; // 0x0D18 (0x0004)
[0x0000000000002000] [0x00000008] (CPF_Transient) int32_t PointsScoredThisRound; // 0x0D1C (0x0004)
[0x0000000000002000] (CPF_Transient) class UGFxModal_X* ResetModal; // 0x0D20 (0x0008)
[0x0000040000000000] int32_t ShotAttempt; // 0x0D28 (0x0004)
[0x0000000000002000] (CPF_Transient) int32_t RedoTotal; // 0x0D2C (0x0004)
[0x0000000000002000] (CPF_Transient) int32_t GoalieScore; // 0x0D30 (0x0004)
[0x0000000000002000] (CPF_Transient) uint8_t PlayTestType; // 0x0D34 (0x0001)
[0x0000000000002000] (CPF_Transient) TArray<class AActor*> GoalMeshBlockers; // 0x0D38 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink) class AActor* GoalMeshBlockerArchetype; // 0x0D48 (0x0008)
[0x0000000000000000] class USaveData_GameEditor_Training_TA* TrainingData; // 0x0D50

```

```

(0x0008) [0x0000000000000000] (CPF_Transient)
struct FName           PrePlaytestState;           // 0x0D58 (0x0008)
[0x0000000000000000] (CPF_Transient)
float                 SaveDelayTime;             // 0x0D60 (0x0004)
[0x0000000000000000]
float                 SaveCooldown;              // 0x0D64 (0x0004)
[0x0000000000000000]
class FString          TrainingFileName;         // 0x0D68 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTrainingEditorNavigationConfig_TA*   TrainingNavigationConfig; // 0x0D78
(0x0008) [0x0000800000000000]
class UTrainingEditorManipulationConfig_TA*  TrainingManipulationConfig; //
0x0D80 (0x0008) [0x0000800000000000]
class UTrainingEditorNavigation_TA*          TrainingNavigator;        // 0x0D88
(0x0008) [0x000100000002000] (CPF_Transient)
class UTrainingEditorMirror_TA*               TrainingMirror;          // 0x0D90 (0x0008)
[0x000100000002000] (CPF_Transient)
struct FScriptDelegate      __EventRoundStarted__Delegate; // 0x0D98
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventRoundTimeChanged__Delegate; // 0x0DB0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventRoundChanged__Delegate; // 0x0DC8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventRoundFinished__Delegate; // 0x0DE0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventPlaytestStarted__Delegate; // 0x0DF8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventUnsavedChanges__Delegate; // 0x0E10
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventSaveResult__Delegate; // 0x0E28
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventRoundAttempted__Delegate; // 0x0E40
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __FindObjectForTraining__Delegate; // 0x0E58
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_TrainingEditor_TA");
}

return uClassPointer;
};

void OnLoadingMovieClosed();
void HandlePrimaryPlayerChange(class ULocalPlayer* OldPrimary, class ULocalPlayer* NewPrimary);
void TagHistoryChanges();
void MarkAsDirty();

```

```
void ForceTagHistoryChanges();
void SetTracedCrosshairActor(class APlayerController_TA* PC, class AActor* NewActor);
void HandleBallWentThruRing(class ABall_GameEditor_TA* Ball);
void __GameEvent_TrainingEditor_TA__OnInit_0x1(class UGameplaySettingsSave_TA* Settings);
bool AllowDynamicCrowd();
void BroadcastGoMessage();
bool IsEditing();
bool IsPlaytesting();
int32_t GetTotalRounds();
uint8_t GetDifficulty();
uint8_t GetTrainingType();
bool DestroyBall(class ABall_TA* Ball);
void Save();
void HandleSaveResult(struct FSaveObjectResult Result);
void SavelInternal();
void OnTrainingModeLoaded();
void DuplicateRound(int32_t Index);
void SetMetaData(class FString NewName, uint8_t NewType, uint8_t NewDifficulty,
TArray<int32_t> NewTags);
void ReorderRound(int32_t FromIndex, int32_t ToIndex);
void SetRoundTimeLimit(float NewRoundTime);
bool HandleNextGame();
void ResetBalls();
class UTrainingEditorMetrics_TA* GetTrainingMetrics();
void Load(class FString SaveName, class APlayerController_TA* PC);
class AActor* GetFirstFoundArchetype(struct FName ArchetypeName);
void SwapGoals();
void DestroyGoalMeshBlockers();
void UpdateGoalMeshBlocker();
int32_t GetScore();
void RemovePointsFromScore(int32_t PointsToRemove, int32_t TeamIndex);
bool ShowScorerGoalMessage();
void OnResetRoundCancel(class UGFxModal_X* Modal);
void OnResetRoundConfirm(class UGFxModal_X* Modal);
void ShowResetRoundMessage();
bool ShowPenaltyMessage();
int32_t GetPlayerTeamNumber();
int32_t GetBallGoalScoreNumber();
void RemoveAllPointsFromScore(int32_t TeamIndex);
bool IncrementRound(unsigned long bLoop);
int32_t GetStoredArchetypeCount(int32_t RoundNumber, class UClass* ClassType);
void SetupInitialPawnOrientation(class APlayerController* NewPlayer);
void LoadRoundData(struct FEditorRoundData Round, class UClass* ClassType);
void LoadRound(struct FEditorRoundData Round);
struct FVector AdjustToFloorLocation(struct FVector TraceStart, struct FVector CollisionExtent);
class AActor* SpawnArchetypeAtAndAdjustToFloor(class AActor* Archetype, struct FVector
SpawnLocation, struct FRotator SpawnRotation);
void SpawnBallAndStartPointAt(struct FVector BallSpawnLocation, struct FRotator
BallSpawnRotation, struct FVector StartPointSpawnLocation, struct FRotator
StartPointSpawnRotation);
void SetupDefaultRound();
void SwitchToRoundNumber(int32_t RoundNumber, unsigned long BackupCurrentRound);
void SetRoundByNumber(int32_t RoundNumber, struct FEditorRoundData RoundData);
void DeleteRound(int32_t RoundToDelete);
```

```

struct FEditorRoundData GetCurrentRoundData();
static bool CanPlayTestRound(struct FEditorRoundData Round, struct FScriptDelegate
FindObjectCallback);
void CleanupRoundActors();
void EndTraining();
void RestartPlayTest();
void EndPlayTest();
void StartPlayTest(uint8_t InPlayTestType);
bool CanPlayTestRoundNumber(int32_t RoundNumber);
void SetUnsavedChanges(unsigned long bInUnsavedChanges);
void UpdateActiveRoundData();
void StartAtRoundNumber(int32_t NewRoundNumber);
void StartNewRound();
bool IsValidRoundIndex(int32_t ArrayIndex);
void HandleTrainingGameSpeedChange(float NewSpeed);
void HandlePrimaryGameplaySettingsSave(class UGameplaySettingsSave_TA* Settings);
void AddLocalPlayer(class APlayerController_TA* Player);
struct FName GetMatchTypeName();
void OnInit();
void eventDestroyed();
void eventPostBeginPlay();
class UObject* FindObjectForTraining(class FString ObjectName, class UClass* ObjectClass,
unsigned long MayFail);
void EventRoundAttempted(int32_t RoundNumber, uint8_t NewAttempt);
void EventSaveResult(unsigned long bSuccess);
void EventUnsavedChanges(unsigned long bOutUnsavedChanges);
void EventPlaytestStarted(class AGameEvent_TrainingEditor_TA* GamEvent);
void EventRoundFinished(class AGameEvent_TrainingEditor_TA* GamEvent);
void EventRoundChanged(class AGameEvent_TrainingEditor_TA* GamEvent);
void EventRoundTimeChanged(class AGameEvent_TrainingEditor_TA* GameEvent);
void EventRoundStarted(class AGameEvent_TrainingEditor_TA* GameEvent);
};

// Class TAGame.__GameEvent_Tutorial_FreePlay_TA__HandleHitGoal_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEvent_Tutorial_FreePlay_TA__HandleHitGoal_0x1 : public UObject
{
public:
class UGoal_TA* Goal; // 0x0060 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEvent_Tutorial_FreePlay_TA__HandleHitGoal_0x1");
}

return uClassPointer;
};

```

```

void __GameEvent_Tutorial_FreePlay_TA__HandleHitGoal_0x1(class APlayerController_TA* PC);
};

// Class TAGame.GameEvent_Tutorial_TA
// 0x0240 (0x0CA0 - 0x0EE0)
class AGameEvent_Tutorial_TA : public AGameEvent_Soccar_TA
{
public:
    struct FVector           TotalFieldExtent;          // 0x0CA0 (0x000C)
    [0x0000000000000002] (CPF_Const)
    int32_t                  TeamNum;                 // 0x0CAC (0x0004)
    [0x0000000000000000]
    int32_t                  BallGoalNum;            // 0x0CB0 (0x0004)
    [0x0000000000000000]
    unsigned long             bOnlyScoreInBallGoalNum : 1; // 0x0CB4 (0x0004)
    [0x0000000000000000] [0x00000001]
    unsigned long             bRedoRound : 1;           // 0x0CB4 (0x0004)
    [0x0000000000000000] [0x00000002]
    unsigned long             bAllowSuperBoost : 1;      // 0x0CB4 (0x0004)
    [0x0000000000000000] [0x00000004]
    unsigned long             bDisplayedRedoPenaltyMessage : 1; // 0x0CB4
    (0x0004) [0x0000004000000000] [0x00000008]
    unsigned long             bShowBoostMeter : 1;         // 0x0CB4 (0x0004)
    [0x0000000000000000] [0x00000010]
    uint8_t                  Difficulty;              // 0x0CB8 (0x0001)
    [0x0000000000000000]
    uint8_t                  DebugRotationType;        // 0x0CB9 (0x0001)
    [0x0000000000000000]
    struct FDifficultyInfo   DifficultyList[0x3];       // 0x0CC0 (0x0078)
    [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<int32_t>          WaveIndexesLeft;          // 0x0D38 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)
    float                    GoalDepth;                // 0x0D48 (0x0004)
    [0x0000000000000000]
    int32_t                  GameEventRounds;          // 0x0D4C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                    EventStartTime;           // 0x0D50 (0x0004)
    [0x0000000000000000]
    struct FVector           BallInitialVelocity;       // 0x0D54 (0x000C)
    [0x0000000000000000]
    int32_t                  SpawnIndexTypeOverride;     // 0x0D60 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class ACannon_TA*        Cannon;                  // 0x0D68 (0x0008)
    [0x0000000000000000]
    int32_t                  WaveIndex;                // 0x0D70 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    int32_t                  WaveSpawnCount;           // 0x0D74 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    int32_t                  RandomSpawnIndex;         // 0x0D78 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    class UTutorial_TA*      Tutorial;                // 0x0D80 (0x0008)
    [0x0000000000000000]
    class FString             StartMessageArchetype;    // 0x0D88 (0x0010)

```

```

[0x0000080000400002] (CPF_Const | CPF_NeedCtorLink)
struct FVector BallSpawnLocation; // 0x0D98 (0x000C)
[0x0000000000000000]
TArray<struct FCarSpawnData> CarSpawns; // 0x0DA8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t PointsScoredThisRound; // 0x0DB8 (0x0004)
[0x0000000000000000]
int32_t BallSpawnCount; // 0x0DBC (0x0004)
[0x0000000000000000]
float BallBounceScale; // 0x0DC0 (0x0004)
[0x0000000000000000]
class ACannon_TA* CannonArchetype; // 0x0DC8 (0x0008)
[0x0000000000000000]
class ADynamicSpawnPoint_TA* DynamicSpawnPoint; // 0x0DD0
(0x0008) [0x0000000000002000] (CPF_Transient)
int32_t CurrentDebugStepX; // 0x0DD8 (0x0004)
[0x0000000000000000]
int32_t CurrentDebugStepY; // 0x0DDC (0x0004)
[0x0000000000000000]
int32_t CurrentDebugStepZ; // 0x0DE0 (0x0004)
[0x0000000000000000]
int32_t RedoCount; // 0x0DE4 (0x0004)
[0x0000004000000000]
int32_t RedoTotal; // 0x0DE8 (0x0004)
[0x0000004000000000]
class UGFxModal_X* ResetModal; // 0x0DF0 (0x0008)
[0x0000004000000000]
struct FFieldSpawnInfo BallSpawnInfo; // 0x0DF8 (0x003C)
[0x0000000000000000]
struct FScriptDelegate __EventRoundFinished__Delegate; // 0xE38
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventVehicleSetup__Delegate; // 0xE50
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventScoreUpdated__Delegate; // 0xE68
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventDifficultyUpdated__Delegate; // 0xE80
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventShowBoostMeterChanged__Delegate; // 0xE98
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventUseActionsChanged__Delegate; // 0xEB0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTutorialTipChanged__Delegate; // 0xEC8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Tutorial_TA");
}
}

```

```
return uClassPointer;
};

void InitIntro();
void OnLoadingMovieClosed();
void StartTimers();
void UpdateMVP();
bool AllowDynamicCrowd();
void SetTutorialTip(class FString NewTip);
void SetShowBoostMeter(unsigned long bShow);
struct FName GetDesiredSoundState();
struct FVector GetDebugSpawnLocation(struct FFieldSpawnInfo SpawnInfo);
float GetStepLoc(int32_t Steps, float TotalDist, unsigned long bIncrement, int32_t& Out_CurrentStep);
struct FVector GetDebugLocationInExtent(struct FVector Extent);
void InitDebugSetup(class ACar_TA* Car);
void SkipTutorial();
void UpdateBotCount();
void InitMutators();
bool IsPrimaryPlayer(class ACar_TA* Car);
bool IsPerfectRound();
bool CanAwardPoints();
void OnResetRoundCancel(class UGFxModal_X* Modal);
void OnResetRoundConfirm(class UGFxModal_X* Modal);
void ShowResetRoundMessage();
void Destroyed();
bool EndTutorial();
bool StartTutorial(class APlayerController_TA* PC, class FString TutorialName, class USequenceAction* KNode, unsigned long bSkipFadeIn);
void CommitRedoRound();
void RedoRound();
bool CanRedoRound();
void StartNewRound();
void SaveLocalPlayerStats();
class ATeam_TA* GetWinningTeam();
void CleanupRoundActors();
struct FVector DetermineBallInitialVelocity(struct FVector BallSpawnLoc, class ACar_TA* Car, struct FFieldSpawnInfo SpawnInfo);
bool CanQueSaveReplay();
void ResetBalls();
struct FName GetScoreStatus(uint8_t InTeamNum);
int32_t GetScore();
void StartRound();
int32_t GetGameEventRounds();
int32_t GetTotalRounds();
void ResetRoundTime();
void OnPlayerRestarted(class ACar_TA* PlayerCar);
void ResetOrientation(class AController* Controller, struct FVector Loc, struct FRotator Rot);
struct FVector GetBallInitialVelocity(struct FVector BallSpawnLoc, class ACar_TA* Car, struct FFieldSpawnInfo SpawnInfo);
uint8_t GetTrainingType();
class FString GetDifficulty();
class FString GetTrainingName();
void EndGame();
```

```
class ATeam_TA* PickTeam(class AController* C);
struct FVector GetRandomLocationInExtent(struct FVector Extent);
bool Chance(int32_t Chances);
int32_t GetOppositeTeamNum();
struct FVector GetDirectionToGoal(int32_t GoalNum);
bool IsGameEventComplete();
struct FVector ClampPointToExtent(struct FVector ExtentCenter, struct FVector Point, struct FVector Extent);
struct FFieldSpawnInfo GetRandomBallSpawnInfo();
struct FFieldSpawnInfo GetRandomPlayerSpawnInfo();
struct FVector PredictInitialVelocity(struct FVector StartLoc, struct FVector EndLoc, float Z);
class ABall_TA* GetGameBall();
class ACar_TA* GetGamePawn();
void ResetGameEvent();
void CheckForReset();
static float GetGoalViewWidth(class UGoal_TA* Goal, struct FVector ViewerLoc);
static bool IsBallMovingTowardsGoal(class UGoal_TA* Goal, class ABall_TA* Ball, float MinVelocityForDestroy, float InGoalDepth);
void SetGoalDepth();
int32_t GetShuffledSpawnIndex();
struct FCarSpawnData GetCarSpawnData(class AController* Controller);
void AddToCarSpawnData(class AController* Controller, struct FVector SpawnLocation, struct FRotator SpawnRotation);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct FRotator& out_Rotation);
struct FRotator GetCarSpawnRotation(struct FFieldSpawnInfo SpawnInfo, struct FVector CarSpawnLocation);
struct FVector GetCarSpawnLocation(struct FFieldSpawnInfo SpawnInfo);
void DestroyCannon();
void SetCannonOrientation(struct FVector NewLocation, struct FRotator NewRotation);
class ABall_TA* SpawnBall(struct FVector SpawnLoc, unsigned long bWake, unsigned long bSpawnCannon, class FString BallArch);
void InitBallEffects();
void InitBallVelocity();
struct FVector GetRandomGoalAimLocation(int32_t InTeamNum, struct FVector BallLoc);
static struct FVector GetGoalExtent(class UGoal_TA* Goal);
void SetBallVelocity(struct FVector InitialVelocity, class ABall_TA* Ball);
struct FVector GetSpawnLocation(struct FFieldSpawnInfo SpawnInfo);
struct FVector GetMaxFieldExtent(struct FFieldSpawnInfo SpawnInfo);
void InitGameSetup(class ACar_TA* Car);
bool ShouldAllowSuperBoost();
void OnVehicleSetup(class ACar_TA* Car);
void HandleVehicleSetup(class ACar_TA* Car);
void InitCrowdManager();
void HandleScoreUpdated(class ATeam_TA* Team);
void SetDifficulty(int32_t InDifficulty);
void UpdateStats();
void AddLocalPlayer(class APlayerController_TA* Player);
void HandlePlayerResetTraining(class AGameEvent_TA* GameEvent);
void OnInit();
void EventTutorialTipChanged(class AGameEvent_Tutorial_TA* GameEvent, class FString NewTip);
void EventUseActionsChanged(class AGameEvent_Tutorial_TA* GameEvent);
void EventShowBoostMeterChanged(class AGameEvent_Tutorial_TA* GameEvent);
```

```

void EventDifficultyUpdated(class AGameEvent_Tutorial_TA* GameEvent);
void EventScoreUpdated(class AGameEvent_Tutorial_TA* GameEvent);
void EventVehicleSetup(class AGameEvent_Tutorial_TA* GameEvent, class ACar_TA* Car);
void EventRoundFinished(class AGameEvent_Tutorial_TA* GameEvent);
};

// Class TAGame.GameEvent_Tutorial_FreePlay_TA
// 0x0000 (0x0EE0 - 0x0EE0)
class AGameEvent_Tutorial_FreePlay_TA : public AGameEvent_Tutorial_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Tutorial_FreePlay_TA");
}

return uClassPointer;
};

void Destroyed();
void ResetBalls();
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct
FRotator& out_Rotation);
void InitMutators();
void ResetGameEvent();
void CheckForReset();
void OnVehicleSetup(class ACar_TA* Car);
void InitGameSetup(class ACar_TA* Car);
};

// Class TAGame._GameEventProductAssetCache_TA__HandleCarAssetsLoaded_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEventProductAssetCache_TA__HandleCarAssetsLoaded_0x1 : public UObject
{
public:
class UProductLoader_TA* Loader; // 0x0060 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GameEventProductAssetCache_TA__HandleCarAssetsLoaded_0x1");
}
}

```

```

return uClassPointer;
};

bool __GameEventProductAssetCache_TA__HandleCarAssetsLoaded_0x1(class
UPriAssetCache_TA* C);
};

// Class TAGame.PriAssetCache_TA
// 0x0018 (0x0060 - 0x0078)
class UPriAssetCache_TA : public UObject
{
public:
class APRI_TA*          PRI;           // 0x0060 (0x0008)
[0x0000000000000000]
TArray<int32_t>          Products;      // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PriAssetCache_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameEventProductAssetCache_TA
// 0x001C (0x00A4 - 0x00C0)
class UGameEventProductAssetCache_TA : public UActorComponent_X
{
public:
class UProductAssetCache_TA*      AssetCache;    // 0x00A8
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<class UPriAssetCache_TA*>   PriCache;     // 0x00B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEventProductAssetCache_TA");
}

return uClassPointer;
}

```

```

};

int32_t __GameEventProductAssetCache_TA__HandleCarAssetsLoaded_0x2(class UProductAsset_TA* P);
void __GameEventProductAssetCache_TA__RefreshCache_0x1(class UPriAssetCache_TA* C);
void RefreshCache();
void RefreshCacheDelayed();
void HandleCarAssetsLoaded(class UProductLoader_TA* Loader);
void HandleCarSet(class APRI_TA* PRI);
void HandlePlayerRemoved(class AGameEvent_TA* G, class APRI_TA* PRI);
void HandlePlayerAdded(class AGameEvent_TA* G, class APRI_TA* PRI);
void eventDetached();
void eventAttached();
};

// Class TAGame.__GameEventProductAssetCache_TA__HandlePlayerAdded_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEventProductAssetCache_TA__HandlePlayerAdded_0x1 : public UObject
{
public:
    class APRI_TA*          PRI;           // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GameEventProductAssetCache_TA__HandlePlayerAdded_0x1");
        }

        return uClassPointer;
    }

    bool __GameEventProductAssetCache_TA__HandlePlayerAdded_0x1(class UPriAssetCache_TA*
C);
};

// Class TAGame.__GameEventProductAssetCache_TA__HandlePlayerRemoved_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameEventProductAssetCache_TA__HandlePlayerRemoved_0x1 : public UObject
{
public:
    class APRI_TA*          PRI;           // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameEventProductAssetCache_TA__HandlePlayerRemoved_0x1");
}

return uClassPointer;
};

bool __GameEventProductAssetCache_TA__HandlePlayerRemoved_0x1(class
UPriAssetCache_TA* C);
};

// Class TAGame.__GameInfo_GFxMenu_TA__Destroyed_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameInfo_GFxMenu_TA__Destroyed_0x1 : public UObject
{
public:
class UGFxEngine_TA*           GFxEngine;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameInfo_GFxMenu_TA__Destroyed_0x1");
}

return uClassPointer;
};

void __GameInfo_GFxMenu_TA__Destroyed_0x1(class UGFxShell_X* Shell);
};

// Class TAGame.GameInfoBase_TA
// 0x0010 (0x04C8 - 0x04D8)
class AGameInfoBase_TA : public AGameInfo_X
{
public:
TArray<class UGFxSoundPack_X*>      SoundPacks;        // 0x04C8
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfoBase_TA");
}
}

```

```

}

return uClassPointer;
};

static UClass* eventSetGameType(class FString MapName, class FString Options, class
FString Portal);
};

// Class TAGame.GameInfo_GFxMenu_TA
// 0x00A0 (0x04D8 - 0x0578)
class AGameInfo_GFxMenu_TA : public AGameInfoBase_TA
{
public:
class UCarPreviewSet_TA*           CarPreviewSet;           // 0x04D8 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UPremiumGaragePreviewSet_TA*   PremiumPreviewSet;   // 0x04E0
(0x0008) [0x0000004000002000] (CPF_Transient)
class UExplosionPreviewer_TA*       ExplosionPreviewer; // 0x04E8
(0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
CPF_EditInline)
class UPostMatchCelebrationPreviewer_TA* PostMatchPreviewer; // 
0x04F0 (0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
CPF_EditInline)
class UMusicStingersPreviewer_TA*     MusicStingersPreviewer; // 0x04F8
(0x0008) [0x0000004000000000]
class UMenuSequencer_TA*            MenuSequencer;        // 0x0500
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_EditInline)
class FString                      UIStateNames[0x2];    // 0x0508 (0x0020)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FName                        UISoundState_MainMenu; // 0x0528 (0x0008)
[0x0000000000000002] (CPF_Const)
TArray<class UCinematicIntroSequence_TA*> CinematicSequences; // 
0x0530 (0x0010) [0x000100000400002] (CPF_Const | CPF_NeedCtorLink)
class UEnableOnlineSave_X*          OnlineSaveEnable;      // 0x0540
(0x0008) [0x0000000000000000]
struct FScriptDelegate              __EventIntroCarsInPosition__Delegate; // 0x0548
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              __EventUIStateChanged__Delegate; // 0x0560
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_GFxMenu_TA");
}

return uClassPointer;
};

```

```

bool AllowPausing(class APlayerController* PC);
bool UIStatelsActive(class UGFxShell_TA* Shell, uint8_t UIState);
void TriggerUIStateChangeEvents(uint8_t UIState);
void SetUIState(class ULocalPlayer_TA* Player, uint8_t UIState);
void HandlePrimaryPlayerIdChanged(class UOnlineGameAccount_X* Account, struct
FUniqueNetId PlayerID);
void SetUIMenuState(uint8_t MenuState);
void HandlePrimaryPlayerSet(class APlayerController_Menu_TA* PC);
bool ShouldSkipTitleScreen(class ULocalPlayer_TA* LocalPlayer);
void InitShell(class APlayerController_Menu_TA* PC);
void GenericPlayerInitialization(class AController* C);
bool CanStartMatch();
void RestartPlayer(class AController* NewPlayer);
void eventDestroyed();
void InitMenuSequencer();
void InitPreviews();
void eventPreBeginPlay();
void EventUIStateChanged(class AGameInfo_GFxFxMenu_TA* GameInfoMenu, uint8_t UIState);
void EventIntroCarsInPosition(class AGameInfo_GFxFxMenu_TA* GameInfoMenu);
};

// Class TAGame.__GameInfo_Soccar_TA__QueueShutdown_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameInfo_Soccar_TA__QueueShutdown_0x1 : public UObject
{
public:
class UOnlineGameDedicatedServer_X* DedicatedServer; // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameInfo_Soccar_TA__QueueShutdown_0x1");
}

return uClassPointer;
};

void __GameInfo_Soccar_TA__QueueShutdown_0x1(class AGameEvent_Soccar_TA* _);

// Class TAGame.GameInfo_TA
// 0x0068 (0x04D8 - 0x0540)
class AGameInfo_TA : public AGameInfoBase_TA
{
public:
TArray<struct FPostMatchCelebrationData> PostMatchCelebrations; // 0x04D8 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

unsigned long          bShutdownQueued : 1;           // 0x04E8 (0x0004)
[0x0000008000002000] [0x00000001] (CPF_Transient)
unsigned long          bShuttingDown : 1;           // 0x04E8 (0x0004)
[0x0000008000002000] [0x00000002] (CPF_Transient)
class AGameEvent_TA*   CurrentGame;                // 0x04F0 (0x0008)
[0x0000008000002000] (CPF_Transient)
struct FServerConnectionInfo ReplacementServerInfo; // 0x04F8
(0x0030) [0x0000008000400000] (CPF_NeedCtorLink)
struct FScriptDelegate   __EventGameInitialized__Delegate; // 0x0528
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_TA");
}

return uClassPointer;
};

struct FName RollCelebrationAnim(struct FName InCelebrationName, unsigned long bIsMVP);
void RegisterCelebration(struct FName InCelebrationName, struct FName InMVPAnim,
TArray<struct FName> InRegularAnims);
void RemoveBot(int32_t Index);
void SpawnDummyBot(struct FName BodyName);
void PlayerResetTraining();
void StartGameEvent(struct FGameEventData GameData);
bool AllowPausing(class APlayerController* PC);
class UGameMode_TA* GetGameMode();
bool AtCapacity(unsigned long bSpectator);
void HandleTimeDilationChanged(class AWorldInfo* WI);
void RestartLevel();
void NotifyClientsServerShutdown();
void StartShutdown();
void QueueShutdown(class FString Reason);
void ShutdownGame();
struct FVector GetGoodSpawnPointNear(class AActor* A, struct FVector PossibleLoc);
void RestartPlayer(class AController* NewPlayer);
void GenericPlayerInitialization(class AController* C);
void eventAddDefaultInventory(class APawn* P);
class FString GetFakePlayerName(class AController* Other);
class APlayerController* eventLogin(class FString Portal, class FString Options, struct
FUniqueNetId UniqueId, class FString& ErrorMessage);
void eventPreLogin(class FString Options, class FString Address, struct FUniqueNetId UniqueId,
unsigned long bSupportsAuth, class FString& ErrorMessage);
void eventInitGame(class FString Options, class FString& ErrorMessage);
class UIOnlineGameHost_X* GetOnlineGameHost();
void EventGameInitialized(class AGameInfo_TA* Game);
};

```

```

// Class TAGame.GameInfo_Soccar_TA
// 0x0010 (0x0540 - 0x0550)
class AGameInfo_Soccar_TA : public AGameInfo_TA
{
public:
TArray<struct FGameEventData>           GameEventArchetypes;          // 0x0540
(0x0010) [0x000000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Soccar_TA");
}

return uClassPointer;
};

bool HasMoreThanOneRemotePlayer();
void ForceInactive();
void KickAllPlayers(class FString Reason);
void RestartLevel();
void RestartPlayer(class AController* aPlayer);
bool PlayerCanRestart(class APlayerController* aPlayer);
void HandleNewDedicatedServerForPlayers(class UConnectionInfoMessage_X* Message);
void QueueShutdown(class FString Reason);
void RestartGameEvent();
bool IsGameEventType(class FString LaunchOption, class UObject* Archetype);
void HandleMainEventDestroyed(class AGameEvent_TA* GameEvent);
void OnCreateCurrentGame();
void InitGameEvent(class AGameEvent_TA* GameEventArchetype, unsigned long bAddAllPlayers);
void StartGameEvent(struct FGameEventData GameData);
void StartGameEventFromLaunchOption(class FString LaunchOption);
void StartDefaultGameEvent();
void eventPostLogin(class APlayerController* NewPlayer);
class APlayerController* eventLogin(class FString Portal, class FString Options, struct FUniqueNetId UniqueId, class FString& ErrorMessage);
void eventPreLogin(class FString Options, class FString Address, struct FUniqueNetId UniqueId, unsigned long bSupportsAuth, class FString& ErrorMessage);
void eventPostBeginPlay();
};

// Class TAGame.__GameInfo_TA__RegisterCelebration_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameInfo_TA__RegisterCelebration_0x1 : public UObject
{
public:
struct FName           InCelebrationName;          // 0x0060 (0x0008)
[0x0000000000000000]

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GameInfo_TA__RegisterCelebration_0x1");
}

return uClassPointer;
};

bool __GameInfo_TA__RegisterCelebration_0x1(struct FPostMatchCelebrationData PMC);
};

// Class TAGame.__GameInfo_TA__RollCelebrationAnim_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameInfo_TA__RollCelebrationAnim_0x1 : public UObject
{
public:
struct FName           InCelebrationName;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameInfo_TA__RollCelebrationAnim_0x1");
}

return uClassPointer;
};

bool __GameInfo_TA__RollCelebrationAnim_0x1(struct FPostMatchCelebrationData PMC);
};

// Class TAGame.__GameObserver_TA__GetCarDataForPRI_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameObserver_TA__GetCarDataForPRI_0x1 : public UObject
{
public:
class APRI_TA*          PRI;                      // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameObserver_TA__GetCarDataForPRI_0x1");
}

return uClassPointer;
};

bool __GameObserver_TA__GetCarDataForPRI_0x1(struct FCarData C);
};

// Class TAGame.NameplateComponent_TA
// 0x003C (0x00A4 - 0x00E0)
class UNameplateComponent_TA : public UActorComponent_X
{
public:
class FString PlayerName; // 0x00A8 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
int32_t Team; // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bIsLocalPlayer : 1; // 0x00BC (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bIsReady : 1; // 0x00BC (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bLocalizeName : 1; // 0x00BC (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
int32_t SpectatorShortcut; // 0x00C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FScriptDelegate __EventPlayerNameSet__Delegate; // 0x00C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NameplateComponent_TA");
}

return uClassPointer;
};

int32_t GetSpectatorShortcut();
bool IsReady();
bool IsLocalPlayer(class APRI_TA* LocalPRI);
int32_t GetPlayerTeam();
class FString GetPlayerName();
class APRI_TA* GetPRI();
void SetPlayerName(class FString InPlayerName);
void eventDetached();
void eventAttached();

```

```
void EventPlayerNameSet(class UNameplateComponent_TA* NameplateComponent);
};

// Class TAGame.__GameShare_TA__NotifyCarPreviewActorAdded_0x1
// 0x0018 (0x0060 - 0x0078)
class U__GameShare_TA__NotifyCarPreviewActorAdded_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x00000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GameShare_TA__NotifyCarPreviewActorAdded_0x1");
        }

        return uClassPointer;
    };

void __GameShare_TA__NotifyCarPreviewActorAdded_0x1(class ACarPreviewActor_TA* C);
};

// Class TAGame.__GameTags_TA__ConvertToGameTags_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GameTags_TA__ConvertToGameTags_0x1 : public UObject
{
public:
    class UObject*                  InOuter;          // 0x0060 (0x0008)
    [0x00000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GameTags_TA__ConvertToGameTags_0x1");
        }

        return uClassPointer;
    };

    class UGameTags_TA* __GameTags_TA__ConvertToGameTags_0x1(class UGameTags_TA* G);
};

// Class TAGame.GameTags_TA
```

```
// 0x0028 (0x0060 - 0x0088)
class UGameTags_TA : public UObject
{
public:
TArray<struct FKeyValuePair>           Tags;           // 0x0060 (0x0010)
[0x0008004000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __Tags__ChangeNotify; // 0x0070 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameTags_TA");
}

return uClassPointer;
};

void __Tags__ChangeNotifyFunc();
static void ConvertToGameTags(class UObject* InOuter, TArray<class UGameTags_TA*>& GameTags, TArray<struct FKeyValuePair>& Pairs);
void RemoveValue(class FString Value);
class FString CreateGameTagsStr();
void SetValue(class FString Key, class FString Value);
class FString GetValue(class FString Key);
void ResetTags();
void InitTags(class FString AllTags);
};

// Class TAGame.__GameViewportClient_TA__AllowTournamentSplitscreen_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__GameViewportClient_TA__AllowTournamentSplitscreen_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerIDCompare;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GameViewportClient_TA__AllowTournamentSplitscreen_0x1");
}

return uClassPointer;
};
```

```

bool __GameViewportClient_TA__AllowTournamentSplitscreen_0x1(struct FTourPlayer P);
};

// Class TAGame.__GFxDATA_AutoTour_TourCard_TA__HandleSubscriptionChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_AutoTour_TourCard_TA__HandleSubscriptionChanged_0x1 : public UObject
{
public:
    uint64_t TournamentID; // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_AutoTour_TourCard_TA__HandleSubscriptionChanged_0x1");
        }
    }

    return uClassPointer;
};

bool __GFxDATA_AutoTour_TourCard_TA__HandleSubscriptionChanged_0x1(class
UTourSettings_TA* S);
};

// Class TAGame.GFxDATA_AutoTour_TourCard_TA
// 0x0060 (0x0094 - 0x00F4)
class UGFxDATA_AutoTour_TourCard_TA : public UGFxDATARow_X
{
public:
    class UOnlineGameTournaments_TA* OnlineGameTournaments; // 0x0098 (0x0008) [0x0001800000000001] (CPF_Edit)
    TArray<class UTourSettings_TA*> Settings; // 0x00A0 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)
    uint64_t StartTime; // 0x00B0 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t LastTournamentTime; // 0x00B8 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    class FString Title; // 0x00C0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    class FString Description; // 0x00D0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    uint64_t ScheduleID; // 0x00E0 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    int32_t TeamSize; // 0x00E8 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint8_t GameMode; // 0x00EC (0x0001)
    [0x0001000040000000] (CPF>EditInlineNotify)
    unsigned long bIsIneligible : 1; // 0x00F0 (0x0004)
}

```

```

[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bCanCheckIn : 1;           // 0x00F0 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long          bRegistered : 1;         // 0x00F0 (0x0004)
[0x0001000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long          bUpdateSkill : 1;        // 0x00F0 (0x0004)
[0x0001000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long          bIsInitialized : 1;      // 0x00F0 (0x0004)
[0x0001000000000000] [0x00000010]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_AutoTour_TourCard_TA");
}

return uClassPointer;
};

bool __GFxData_AutoTour_TourCard_TA__Init_0x1(class UTourSettings_TA* S);
bool __GFxData_AutoTour_TourCard_TA__GetEligibleTourSetting_0x1(class UTourSettings_TA* S);
bool GetHoldsIneligibleScheduleID();
class UTourSettings_TA* GetEligibleTourSetting();
uint64_t GetActiveTourID();
void HandleSubscriptionChanged(class UTourSubscriptions_TA* Subscriptions, uint64_t TournamentID);
void OnCheckInClosed();
void OnCheckInOpen();
void UpdateCanCheckIn();
void UpdateTitle();
void Init(struct FScheduledTournament& Schedule);
};

// Class TAGame.__GFxData_CarRumble_TA__HandlePawnSet_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_CarRumble_TA__HandlePawnSet_0x1 : public UObject
{
public:
class ACar_TA*          NewCar;                // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class"

```

```

TAGame.__GFxData_CarRumble_TA__HandlePawnSet_0x1");
}

return uClassPointer;
};

void __GFxData_CarRumble_TA__HandlePawnSet_0x1();
};

// Class TAGame.GFxData_CarRumble_TA
// 0x0028 (0x0098 - 0x00C0)
class UGFxData_CarRumble_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FRumblePickupData> Items; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t SelectedItem; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long bHasItems : 1; // 0x00AC (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
int32_t ItemCount; // 0x00B0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t PreviewTimeSeconds; // 0x00B4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class ARumblePickups_TA* Pickups; // 0x00B8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_CarRumble_TA");
}

return uClassPointer;
};

void UpdatePreviewSeconds();
void UpdateSelectedItem();
void UpdateItems();
void UpdateHasItems();
void HandleConcurrentItemCountSet();
void HandleRumblePickupsSet(class ACar_TA* Car);
void HandlePawnSet(class APlayerController_X* PC, class APawn* OldPawn, class APawn* NewPawn);
void HandlePlayerControllerSet(class UPlayer* InPlayer);
void eventOnShellSet();
};

// Class TAGame.GFxData_Chat_TA
// 0x00A8 (0x0098 - 0x0140)

```

```

class UGFxData_Chat_TA : public UGFxDataSingleton_X
{
public:
    int32_t           MaxMessages;          // 0x0098 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t           MaxActiveChatTabs;    // 0x009C (0x0004)
    [0x0000000000002000] (CPF_Transient)
    TArray<struct FActiveChatTab> ActiveChatTabs;      // 0x00A0 (0x0010)
    [0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
    TArray<struct FChatPresetMessageGroup> PresetGroups;   // 0x00B0
    (0x0010) [0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
    TArray<struct FChatPresetMessage> PresetMessages;     // 0x00C0
    (0x0010) [0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
    TArray<struct FGFxChatMessage> Messages;            // 0x00D0
    (0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    unsigned long      bPreMatchLobby : 1;                // 0x00E0 (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    uint8_t            LastMessagedCrossPlatformChatState; // 0x00E4
    (0x0001) [0x0000000000000000]
    class UChatHistory_TA* ChatHistory;                  // 0x00E8 (0x0008)
    [0x0000000000000000]
    int32_t           MaxNumGroups;        // 0x00F0 (0x0004)
    [0x0000000000000002] (CPF_Const)
    int32_t           MaxNumMessagesPerGroup; // 0x00F4 (0x0004)
    [0x0000000000000002] (CPF_Const)
    struct FChatPresetGroupOverride PostGameGroupOverride; // 0x00F8
    (0x0020) [0x0000000000040002] (CPF_Const | CPF_NeedCtorLink)
    struct FChatPresetGroupOverride PreGameGroupOverride; // 0x0118
    (0x0020) [0x0000000000040002] (CPF_Const | CPF_NeedCtorLink)
    class UAprilConfig_X* AprilConfig;           // 0x0138 (0x0008)
    [0x0000800000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_Chat_TA");
        }

        return uClassPointer;
    };

    void __GFxData_Chat_TA__OnShellSet_0x1(class UObject* AccountSettings);
    void ClearDistracted();
    void OnOpenPartyChat();
    void OpenPartyChat();
    void OnOpenTeamChat();
    void OpenTeamChat();
    void OnOpenChat();
    void OpenChat();
    TArray<struct FGFxChatMessage> GetMatchChatLog(struct FUniqueNetId Personald);
}

```

```

void RepopulatedChat();
void CloseIndividualChatTab(struct FUniqueNetId Personald);
void RepopulateIndividualChat(struct FUniqueNetId Personald);
void ClearChat();
void ClearChatByType(uint8_t ChatChannel, struct FUniqueNetId Persona);
void OnChatMessage(int32_t Team, class FString PlayerName, class FString Message, uint8_t ChatChannel, unsigned long bLocalPlayer, struct FUniqueNetId SenderId, uint8_t MessageType, class FString TimeStamp);
void HandleMessageSanitizeError(class UError* Error, uint8_t Channel);
void InternalAddMessage(struct FGfxChatMessage& Data);
void AddRepopulatedMessage(struct FGfxChatMessage NewMessage);
void AddPresetMessage(struct FGfxChatMessage NewMessage);
void AddTransientMessage(struct FGfxChatMessage NewMessage);
void AddChatMessage(struct FGfxChatMessage NewMessage);
void DispatchChatMessage(class FString Message, uint8_t ChatChannel, struct FUniqueNetId Recipient);
void EraseChatByType(uint8_t ChatChannel, struct FUniqueNetId Personald);
void HandleReceivedUnfriendNotice(struct FUniqueNetId Personald);
void HandleIndividualChat(class FString InMessage, struct FUniqueNetId SenderId, unsigned long bIsLocal);
void SendPartyChatMessage(class FString Message, unsigned long bCanceled);
void SendTeamChatMessage(class FString Message, unsigned long bCanceled);
void SendGlobalChatMessage(class FString Message, unsigned long bCanceled);
void SendChatChannelMessage(class FString Message, uint8_t ChatChannel, struct FUniqueNetId Recipient);
void SendChatMessage(class FString Message, unsigned long bTeamMessage);
void OnPressChatPreset(int32_t Index);
void SendChatPresetMessage(struct FName MessageId, unsigned long bTeam);
void DisplayFreshCrossPlatformFilterHint(uint8_t NewCrossplayChatState, class UOnlineGameParty_X* Party);
void HandleCrossPlatformChatStateChanged();
void HandlePartyChanged(class UOnlineGameParty_X* Party);
void InitChatTabs();
void UpdateChatGroups(struct FChatPresetGroupOverride GroupOverride);
void HandleQuickChatSaveChanged();
class APlayerController_TA* GetPlayerController();
class AGameEvent_Soccar_TA* GetGameEvent();
void eventOnRemoved();
void HandleGameEnded(class AGameEvent_Soccar_TA* GameEvent);
void HandleStateChanged(class AGameEvent_TA* GameEvent);
void RefreshQuickChat(class AGameEvent_Soccar_TA* GameEvent);
void InitializeQuickChat(class AGameEvent_Soccar_TA* GameEvent, class UProfileQuickChatSave_TA* QuickChatSave);
static void OnQuickChatAdded(class UGfxData_Chat_TA* GFxChat, class AGameEvent_Soccar_TA* GameEvent, class UProfileQuickChatSave_TA* QuickChatSave);
void OnShellSet();
};

// Class TAGame.__GFxDATA_Chat_TA__AddChatMessage_0x1
// 0x00890 (0x0060 - 0x00EF0)
class U_GFxDATA_Chat_TA__AddChatMessage_0x1 : public UObject
{
public:
    struct FGfxChatMessage           NewMessage;          // 0x0060 (0x00890)

```

[0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._GFxData_Chat_TA__AddChatMessage_0x1");  
}  
  
return uClassPointer;  
};  
  
void __GFxData_Chat_TA__AddChatMessage_0x2(class FString _, class UError* Error);  
void __GFxData_Chat_TA__AddChatMessage_0x1(class FString _, class FString Sanitized);  
};  
  
// Class TAGame.ChatHistory_TA  
// 0x0014 (0x0060 - 0x0074)  
class UChatHistory_TA : public UObject  
{  
public:  
TArray<class UStoredChatData_TA*>           StoredChat;           // 0x0060  
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)  
int32_t                         MaxNumHistories;          // 0x0070 (0x0004)  
[0x0000000000002000] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ChatHistory_TA");  
}  
  
return uClassPointer;  
};  
  
bool __ChatHistory_TA__FindHistory_0x1(class UStoredChatData_TA* N);  
int32_t __ChatHistory_TA__StoreMessageInternal_0x1(class UStoredChatData_TA* L, class  
UStoredChatData_TA* R);  
void RepopulateChatFromHistory(class UGFxData_Chat_TA* GFxChat, uint8_t Channel, struct  
FUniqueNetId SenderId);  
void ClearHistory(uint8_t Channel, struct FUniqueNetId SenderId);  
void StoreMessageInternal(class UStoredChatData_TA* ChatInfo, struct FGFxChatMessage&  
Data);  
class UStoredChatData_TA* CreateNewHistory(uint8_t Channel, struct FUniqueNetId SenderId);  
class UStoredChatData_TA* FindHistory(uint8_t Channel, struct FUniqueNetId SenderId);  
class UStoredChatData_TA* GetOrCreateHistory(uint8_t Channel, struct FUniqueNetId SenderId);
```

```
void StoreMessage(struct FGFxChatMessage& Data);
void ActivateStoredChat(class UStoredChatData_TA* StoredChatHistory);
void DeactivateStoredChatForPersona(struct FUniqueNetId Personald);
bool CanStoreMessage(uint8_t ChannelMessage);
};

// Class TAGame.__GFxDATA_CHAT_TA__AddPresetMessage_0x1
// 0x00890 (0x0060 - 0x00EF0)
class U_GFxDATA_CHAT_TA__AddPresetMessage_0x1 : public UObject
{
public:
    struct FGFxChatMessage           NewMessage;           // 0x0060 (0x00890)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_CHAT_TA__AddPresetMessage_0x1");
        }

        return uClassPointer;
    };

    bool __GFxDATA_CHAT_TA__AddPresetMessage_0x1(struct FQuickChatOverridePair QC);
};

// Class TAGame.__GFxDATA_CHAT_TA__GetMatchChatLog_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxDATA_CHAT_TA__GetMatchChatLog_0x1 : public UObject
{
public:
    struct FUniqueNetId           Personald;           // 0x0060 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_CHAT_TA__GetMatchChatLog_0x1");
        }

        return uClassPointer;
    };

    bool __GFxDATA_CHAT_TA__GetMatchChatLog_0x1(struct FGFxChatMessage msg);
```

```

};

// Class TAGame.__GFxData_ClubDetails_TA__UpdateClubMembers_0x1
// 0x00A0 (0x0060 - 0x0100)
class U_GFxData_ClubDetails_TA__UpdateClubMembers_0x1 : public UObject
{
public:
    struct FClubMember           Member;           // 0x0060 (0x00A0)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ClubDetails_TA__UpdateClubMembers_0x1");
        }

        return uClassPointer;
    };

    bool __GFxData_ClubDetails_TA__UpdateClubMembers_0x1(class UGFxData_ClubMember_TA*
D);
};

// Class TAGame.GFxData_ClubMember_TA
// 0x0094 (0x0094 - 0x0128)
class UGFxData_ClubMember_TA : public UGFxDataRow_X
{
public:
    uint64_t           ClubID;           // 0x0098 (0x0008)
    [0x0001000040000000] (CPF_EditInlineNotify)
    struct FUniqueNetId           MemberId;           // 0x00A0 (0x0048)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    class FString           MemberName;           // 0x00E8 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    class UTexture*           Avatar;           // 0x00F8 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    class FString           UnsanitizedName;           // 0x0100 (0x0010)
    [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
    class UOnlineSubsystem*           OnlineSubsystem;           // 0x0110 (0x0008)
    [0x0001800000000000]
    class UOnlineGameWordFilter_X*           WordFilter;           // 0x0118 (0x0008)
    [0x0001800000000000]
    class UOnlineGame_X*           OnlineGame;           // 0x0120 (0x0008)
    [0x0001800000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClubMember_TA");
}

return uClassPointer;
};

void HandleGetAvatar(struct FUniqueNetId InPlayerId, class UTexture* InAvatar, class FString OnlinePlayerName);
void UpdateAvatar();
void HandleSanitizedName(class FString Original, class FString Sanitized);
void UpdateName();
void SetMember(uint64_t InClubID, struct FUniqueNetId InID, class FString InName);
};

// Class TAGame.GFxData_ClubDetails_TA
// 0x00BC (0x0094 - 0x0150)
class UGFxData_ClubDetails_TA : public UGFxDataRow_X
{
public:
    uint64_t ClubID; // 0x0098 (0x0008)
    [0x0001000040000000] (CPF_EditInlineNotify)
    class FString ClubName; // 0x00A0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    class FString ClubTag; // 0x00B0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    class FString ClubMotD; // 0x00C0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    int32_t PrimaryColorID; // 0x00D0 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    int32_t AccentColorID; // 0x00D4 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    struct FUniqueNetId OwnerId; // 0x00D8 (0x0048)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    unsigned long bVerified : 1; // 0x0120 (0x0004)
    [0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
    unsigned long bCrossPlatformFiltered : 1; // 0x0120 (0x0004)
    [0x0001000040000000] [0x00000002] (CPF>EditInlineNotify)
    class UClubDetails_X* ClubDetails; // 0x0128 (0x0008)
    [0x000100000002000] (CPF_Transient)
    TArray<class UGFxData_ClubMember_TA*> ClubMembers; // 0x0130
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    class UOnlineGameWordFilter_X* WordFilter; // 0x0140 (0x0008)
    [0x0001800000000000]
    class UCrossplayConfig_X* CrossplayConfig; // 0x0148 (0x0008)
    [0x0001800000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClubDetails_TA");
}

return uClassPointer;
};

void __GFxData_ClubDetails_TA__OnShellSet_0x1(class UAccountSettingsComponent_TA*
AccountSettings);
bool CanDisplayClubMember(struct FUniqueNetId NetId, unsigned long bEnableCrossplay);
bool CanViewPlatformWithCrossplayDisabled(struct FUniqueNetId Id0, struct FUniqueNetId Id1);
void UpdateClubMembers();
void HandleSanitizedClubMotD(class FString Original, class FString Sanitized);
void HandleSanitizedClubTag(class FString Original, class FString Sanitized);
void HandleSanitizedClubName(class FString Original, class FString Sanitized);
bool CanAllPlayersCommunicateText();
void ResetMotD();
void HandleNumLocalPlayersChanged(class UGameViewportClient_TA* GVC, int32_t
LocalPlayerCount);
void Destroyed();
void SetClubDetails(class UClubDetails_X* InClubDetails);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.__GFxData_Clubs_TA__AcceptClubInvite_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Clubs_TA__AcceptClubInvite_0x1 : public UObject
{
public:
class UClubInviteNotification_TA*           Notification;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Clubs_TA__AcceptClubInvite_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__AcceptClubInvite_0x2();
void __GFxData_Clubs_TA__AcceptClubInvite_0x1(class UGFxClubAction_TA* Action);
};

// Class TAGame.GFxClubAction_TA
// 0x0048 (0x0060 - 0x00A8)

```

```

class UGFxClubAction_TA : public UObject
{
public:
    class UAsyncTask* Task; // 0x0060 (0x0008)
    [0x0001000000000000]
    class FString ErrorMessage; // 0x0068 (0x0010)
    [0x000100000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __StartTaskDelegate__Delegate; // 0x0078
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __EventComplete__Delegate; // 0x0090
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxClubAction_TA");
        }

        return uClassPointer;
    };

    void eventDispose();
    void OnComplete();
    void HandleTaskComplete(class UError* TaskError);
    class UGFxClubAction_TA* Start(struct FScriptDelegate Action);
    class UGFxClubAction_TA* SetError(class FString Message);
    class UGFxClubAction_TA* AssertNotGuest(struct FUniqueNetId PlayerID);
    class UGFxClubAction_TA* Assert(unsigned long expr, class FString Message);
    void EventComplete(class UGFxClubAction_TA* Action);
    class UAsyncTask* StartTaskDelegate();
};

// Class TAGame.GFxData_Clubs_TA
// 0x00F0 (0x0098 - 0x0188)
class UGFxData_Clubs_TA : public UGFxDataSingleton_X
{
public:
    uint64_t LocalClubID; // 0x0098 (0x0008)
    [0x0001000040000000] (CPF_EditInlineNotify)
    unsigned long bClubOwner : 1; // 0x00A0 (0x0004)
    [0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
    unsigned long bSyncedLocalClub : 1; // 0x00A0 (0x0004)
    [0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
    unsigned long bHasSyncedClubInvites : 1; // 0x00A0 (0x0004)
    [0x0001000000002000] [0x00000004] (CPF_Transient)
    class FString LocalClubSyncError; // 0x00A8 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString ExecutingClubAction; // 0x00B8 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class UPsyNet_X* PsyNet; // 0x00C8 (0x0008)
};

```

```

[0x0001800000000000]
class UOnlineClubProvider_X*           ClubProvider;          // 0x00D0 (0x0008)
[0x0001800000000000]
class UClubsConfig_TA*                ClubsConfig;         // 0x00D8 (0x0008)
[0x0001800000000000]
class UOnlineGameParty_X*             Party;              // 0x00E0 (0x0008)
[0x0001800000000000]
class UPersonas_TA*                  PersonasData;        // 0x00E8 (0x0008)
[0x0001804000002000] (CPF_Transient)
class UOnlineClubManager_X*           ClubManager;        // 0x00F0 (0x0008)
[0x0001000000000000]
TArray<class UGFxData_ClubDetails_TA*> AllClubDetails;      // 0x00F8
(0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
class UGFxClubAction_TA*              CurrentClubAction;   // 0x0108 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString                        TagTooShort;         // 0x0110 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                        TagTooLong;          // 0x0120 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                        TagHasInvalidCharacters; // 0x0130 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                        NameTooShort;        // 0x0140 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                        NameTooLong;         // 0x0150 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                        NameHasInvalidCharacters; // 0x0160 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate               __FlashActionHandler__Delegate; // 0x0170
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Clubs_TA");
}

return uClassPointer;
};

void __GFxData_Clubs_TA__InviteToClub_0x2(class UError* TaskError);
void __GFxData_Clubs_TA__SetClubInvites_0x4(class UClubInviteNotification_TA* Notification);
void __GFxData_Clubs_TA__SetClubInvites_0x3(class UClubInviteNotification_TA* Notification);
bool __GFxData_Clubs_TA__SetClubInvites_0x2(class UClubInviteNotification_TA* N);
void __GFxData_Clubs_TA__RejectAllClubInvites_0x2(class URPC_ClearClubInvites_X* RPC);
void __GFxData_Clubs_TA__RejectAllClubInvites_0x1(class URPC_ClearClubInvites_X* RPC);
class UAsyncTask* __GFxData_Clubs_TA__LeaveClub_0x1();
bool PlayerInCabinMode();
bool CanDisbandClub(uint64_t ClubID);
void LeaveClub();
void RemoveClubInvites(uint64_t ClubID);

```

```

void HandleRejectedClubInviteByIDComplete(class UGFxClubAction_TA* Action, uint64_t ClubID);
void HandleRejectedClubInviteComplete(class UGFxClubAction_TA* Action, class UClubInviteNotification_TA* Notification);
void RejectClubInviteByClubID(uint64_t ClubID);
void RejectClubInvite(class UClubInviteNotification_TA* Notification);
void RejectAllClubInvites();
void HandleAcceptedClubInviteComplete(class UGFxClubAction_TA* Action, class UClubInviteNotification_TA* Notification);
void AcceptClubInvite(class UClubInviteNotification_TA* Notification);
void SetClubInvites(TArray<class UClubInvite_X*> Invites);
class UGFxData_ClubInvite_TA* GetGFxClubInvite(uint64_t ClubID);
void SyncClubInvites();
void SetClubOwner(struct FUniqueNetId PlayerID);
void RemoveFromClub(struct FUniqueNetId PlayerID);
void HandleClubInviteFailed(struct FUniqueNetId PlayerID, class UError* TaskError);
void InviteToClub(struct FUniqueNetId PlayerID);
void SetMotD(class FString Text);
void UpdateClubColors(int32_t PrimaryColorID, int32_t AccentColorID);
void UpdateClubName(class FString ClubName, class FString ClubTag);
void ClearLocalClubInvitations();
void HandleCreateClubFinished(class UGFxClubAction_TA* Action);
void CreateClub(class FString ClubName, class FString ClubTag, int32_t PrimaryColorID, int32_t AccentColorID);
class FString GetClubNameError(class FString ClubName);
class FString GetClubTagError(class FString Tag);
class FString StripInvalidCharacters(class FString Text);
void OnClubActionComplete(class FString Action, class FString Error);
void HandleClubActionComplete(class UGFxClubAction_TA* Action);
class UGFxClubAction_TA* CreateClubAction(struct FName ActionName, struct FScriptDelegate Callback);
void OnSyncClubDetails(uint64_t ClubID, class FString ErrorMessage);
void SyncClubDetails(uint64_t ClubID);
void UpdateGFxClubDetails(class UClubDetails_X* ClubDetails);
void HandleClubChanged(class UOnlineClubCache_X* Cache, class UClubDetails_X* ClubDetails);
void HandlePlayerSynced(class UOnlineClubProvider_X* Provider, struct FUniqueNetId PlayerID);
void UpdateLocalClub();
class UOnlineClubManager_X* GetClubManager();
void HandlePsyNetConnection(class UPsyNetConnection_X* C);
void HandleClubInvite(class UPsyNetService_ClubsInvite_TA* Update);
void eventOnRemoved();
void eventOnShellSet();
void FlashActionHandler(class FString Error);
};

// Class TAGame.__GFxData_Clubs_TA__CreateClub_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Clubs_TA__CreateClub_0x1 : public UObject
{
public:
    class UClubSettings_X*           Settings;          // 0x0060 (0x0008)
    [0x0001000000000000]

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Clubs_TA__CreateClub_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__CreateClub_0x1();
};

// Class TAGame.__GFxData_Clubs_TA__InviteToClub_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxData_Clubs_TA__InviteToClub_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;          // 0x0060 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Clubs_TA__InviteToClub_0x1");
}

return uClassPointer;
};

void
__GFxData_Clubs_TA__InviteToClub_0x1__GFxData_Clubs_TA__InviteToClub_0x1_0x1(class
UError* TaskError);
class UAsyncTask* __GFxData_Clubs_TA__InviteToClub_0x1();
};

// Class TAGame.__GFxData_Clubs_TA__RejectClubInvite_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Clubs_TA__RejectClubInvite_0x1 : public UObject
{
public:
class UClubInviteNotification_TA*      Notification;    // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Clubs_TA__RejectClubInvite_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__RejectClubInvite_0x2();
void __GFxData_Clubs_TA__RejectClubInvite_0x1(class UGFxClubAction_TA* Action);
};

// Class TAGame._GFxData_Clubs_TA__RejectClubInviteByClubID_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Clubs_TA__RejectClubInviteByClubID_0x1 : public UObject
{
public:
uint64_t ClubID; // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Clubs_TA__RejectClubInviteByClubID_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__RejectClubInviteByClubID_0x2();
void __GFxData_Clubs_TA__RejectClubInviteByClubID_0x1(class UGFxClubAction_TA* Action);
};

// Class TAGame._GFxData_Clubs_TA__RemoveFromClub_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxData_Clubs_TA__RemoveFromClub_0x1 : public UObject
{
public:
struct FUniqueNetId PlayerID; // 0x0060 (0x0048)
[0x0001000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Clubs_TA__RemoveFromClub_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__RemoveFromClub_0x1();
};

// Class TAGame.GFxData_ClubInvite_TA
// 0x0028 (0x0128 - 0x0150)
class UGFxData_ClubInvite_TA : public UGFxData_ClubMember_TA
{
public:
class FString ClubName; // 0x0128 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ClubTag; // 0x0138 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UClubInvite_X* Invite; // 0x0148 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClubInvite_TA");
}

return uClassPointer;
};

void HandleSanitizedClubTag(class FString Original, class FString Sanitized);
void HandleSanitizedClubName(class FString Original, class FString Sanitized);
void SetInvite(class UClubInvite_X* InInvite);
};

// Class TAGame.__GFxData_Clubs_TA__SetClubOwner_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__GFxData_Clubs_TA__SetClubOwner_0x1 : public UObject
{
public:
struct FUniqueNetId PlayerID; // 0x0060 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Clubs_TA_SetClubOwner_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA_SetClubOwner_0x1();
};

// Class TAGame.__GFxData_Clubs_TA_SetMotD_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxData_Clubs_TA_SetMotD_0x1 : public UObject
{
public:
class FString Text; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Clubs_TA_SetMotD_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA_SetMotD_0x1();
};

// Class TAGame.__GFxData_Clubs_TA_SyncClubDetails_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Clubs_TA_SyncClubDetails_0x1 : public UObject
{
public:
uint64_t ClubID; // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Clubs_TA_SyncClubDetails_0x1");
}
}

```

```
return uClassPointer;
};

void __GFxData_Clubs_TA__SyncClubDetails_0x1(class UError* Error);
};

// Class TAGame.__GFxData_Clubs_TA__UpdateClubColors_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Clubs_TA__UpdateClubColors_0x1 : public UObject
{
public:
int32_t PrimaryColor; // 0x0060 (0x0004)
[0x0001000000000000]
int32_t AccentColor; // 0x0064 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Clubs_TA__UpdateClubColors_0x1");
}

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__UpdateClubColors_0x1();
};

// Class TAGame.__GFxData_Clubs_TA__UpdateClubName_0x1
// 0x0020 (0x0060 - 0x0080)
class U__GFxData_Clubs_TA__UpdateClubName_0x1 : public UObject
{
public:
class FString ClubName; // 0x0060 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
class FString ClubTag; // 0x0070 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Clubs_TA__UpdateClubName_0x1");
}
}
```

```

return uClassPointer;
};

class UAsyncTask* __GFxData_Clubs_TA__UpdateClubName_0x1();
};

// Class TAGame.__GFxData_Clubs_TA__UpdateGFxClubDetails_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Clubs_TA__UpdateGFxClubDetails_0x1 : public UObject
{
public:
    class UClubDetails_X* ClubDetails; // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Clubs_TA__UpdateGFxClubDetails_0x1");
        }
    }

    return uClassPointer;
};

bool __GFxData_Clubs_TA__UpdateGFxClubDetails_0x1(class UGFxData_ClubDetails_TA* D);
};

// Class TAGame.__GFxData_Community_TA__HandleBlogChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Community_TA__HandleBlogChanged_0x1 : public UObject
{
public:
    uint64_t DateNow; // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Community_TA__HandleBlogChanged_0x1");
        }
    }

    return uClassPointer;
};

```

```

bool __GFxData_Community_TA__HandleBlogChanged_0x1(class UBlogTile_X* X);
};

// Class TAGame.GFxData_Community_TA
// 0x0070 (0x0098 - 0x0108)
class UGFxData_Community_TA : public UGFxDataSingleton_X
{
public:
TArray<class UGFxData_BlogTile_TA*> GFxBlogTiles; // 0x0098
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UBlogConfig_X* Config; // 0x00A8 (0x0008)
[0x0000800000000000]
class FString MotD; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UBlogTileCache_TA* Cache; // 0x00C0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UShopNotificationsManager_TA* ShopNotificationsManager; //
0x00C8 (0x0008) [0x0010800000002000] (CPF_Transient)
unsigned long blsNewsPanelExpanded : 1; // 0x00D0 (0x0004)
[0x0008004000002000] [0x00000001] (CPF_Transient)
unsigned long bInitialLoadCompleted : 1; // 0x00D0 (0x0004)
[0x0008004000002000] [0x00000002] (CPF_Transient)
struct FScriptDelegate __blsNewsPanelExpanded_ChangeNotify; // 0x00D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __bInitialLoadCompleted_ChangeNotify; // 0x00F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Community_TA");
}

return uClassPointer;
};

void __GFxData_Community_TA__OnShellSet_0x1(class UBlogTileCache_TA* InCache);
class FString __GFxData_Community_TA__UpdateFromCache_0x1(class UGFxData_BlogTile_TA*
X);
class UGFxData_BlogTile_TA* __GFxData_Community_TA__HandleBlogChanged_0x2(class
UBlogTile_X* BlogTile);
void __bInitialLoadCompleted_ChangeNotifyFunc();
void __blsNewsPanelExpanded_ChangeNotifyFunc();
void HandleBlogChanged();
void UpdateFromCache();
void ClearBlogTileNewInfo(int32_t Row);
void SetIsNewsPanelExpanded(unsigned long blsExpanded);
void SetInitialLoadCompleted(unsigned long bCompleted);
void eventOnShellSet();
};

```

```

// Class TAGame.__GFxData_ContainerDrops_TA__GetMatchingProductSeries_0x1
// 0x0018 (0x0060 - 0x0078)
class U__GFxData_ContainerDrops_TA__GetMatchingProductSeries_0x1 : public UObject
{
public:
int32_t SeriesID; // 0x0060 (0x0004)
[0x0000000000000000]
TArray<class UOnlineProduct_TA*> SeriesProducts; // 0x0068
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ContainerDrops_TA__GetMatchingProductSeries_0x1");
}

return uClassPointer;
};

bool __GFxData_ContainerDrops_TA__GetMatchingProductSeries_0x3(class UOnlineProduct_TA*
Product);
bool __GFxData_ContainerDrops_TA__GetMatchingProductSeries_0x2(class UOnlineProduct_TA*
P);
bool __GFxData_ContainerDrops_TA__GetMatchingProductSeries_0x1(class UOnlineProduct_TA*
P);
};

// Class TAGame.GFxData_ContainerDrops_TA
// 0x0038 (0x0098 - 0x00D0)
class UGFxData_ContainerDrops_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FContainerInfo> ContainerDropList; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UOnlineProductStoreSet_TA* ContainerDropsSet; // 0x00A8
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
CPF_EditInline)
class UAsyncTask* GetDropTableTask; // 0x00B0 (0x0008)
[0x0000000000000000]
struct FScriptDelegate __EventContainersUpdated__Delegate; // 0x00B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ContainerDrops_TA");
}

return uClassPointer;
};

bool __GFxData_ContainerDrops_TA__HandleGetContainerDropTableResponse_0x1(struct
FRPCCContainerDropTable D);
static int32_t GetBlueprintSeriesID(int32_t BlueprintProductID);
class FString GetPossibleContainerDrops(int32_t ProductID);
class FString GetNonPreviewableProductQualities(int32_t ProductID);
TArray<class FString> GetMatchingProductSeries(int32_t SeriesID);
void OnContainersUpdated();
int32_t OnlineProductToHashInt(class UOnlineProduct_TA* OnlineProduct);
struct FContainerInfo ContainerDropsToHashIDs(struct FRPCCContainerDropTable DropTable);
void HandleGetContainerDropTableResponse(class
URPC_MicroTransactions_GetContainerDropTable_TA* RPC);
class UOnlineProduct_TA* CreateTempOnlineProduct(struct FOnlineProductData Data);
void GetContentDrops();
void HandlePsyNetConnected(class UPsyNetConnection_X* C);
void eventOnRemoved();
void eventOnShellSet();
void EventContainersUpdated(class UGFxData_ContainerDrops_TA* ContainerDrops);
};

// Class TAGame.__GFxData_Controls_TA__ClearBinding_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Controls_TA__ClearBinding_0x1 : public UObject
{
public:
struct FName Action; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Controls_TA__ClearBinding_0x1");
}

return uClassPointer;
};

void __GFxData_Controls_TA__ClearBinding_0x2(class UGFxModal_X* _);
void __GFxData_Controls_TA__ClearBinding_0x1(class UGFxModal_X* _);
};

// Class TAGame.GFxData_Controls_TA
// 0x00D0 (0x0098 - 0x0168)

```

```

class UGFxData_Controls_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FLocalizedBindingCategory> Categories; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FLocalizedAction> Actions; // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FUIPlayerBinding> PCBindings; // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FUIPlayerBinding> GamepadBindings; // 0x00C8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UProfile_TA* Profile; // 0x00D8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UProfileGamepadSave_TA* GamepadSave; // 0x00E0
(0x0008) [0x0000000000002000] (CPF_Transient)
TArray<struct FBindingAction> BindingActions; // 0x00E8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FName PendingRecordBindingAction; // 0x00F8 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FName PendingRecordBindingAxis; // 0x0100 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UIInteraction* Interaction; // 0x0108 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UProfilePCSave_TA* ProfilePCSave; // 0x0110 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString CustomPresetOverwriteWarning; // 0x0118 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString ClearButtonBindingTitle; // 0x0128 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString ClearKeybindTitle; // 0x0138 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString ClearButtonBindingDesc; // 0x0148 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString ClearKeybindDesc; // 0x0158 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Controls_TA");
}

return uClassPointer;
};

void __GFxData_Controls_TA__HandleProfileSet_0x3(class UPlayer* P);
void __GFxData_Controls_TA__HandleProfileSet_0x2(class UProfileGamepadSave_TA* Save);
void __GFxData_Controls_TA__HandleProfileSet_0x1(class UProfilePCSave_TA* Save);
void HandleActiveBindingsChanged(class UPlayerInput_X* PlayerInput);
bool IsRemappableType(struct FName Action, uint8_t RemappableType, TArray<struct

```

```

FPlayerBinding>& Bindings);
bool IsRemappable(struct FName Action, TArray<struct FPlayerBinding> Bindings);
void Save();
void ResetToPreset(struct FName PresetName);
void RevertToDefaults();
void OnBindingSet();
void OnRecordBindingCanceled();
struct FUIPlayerBinding MapUIBinding(struct FPlayerBinding Binding);
void InitUIBindings(struct FName PresetName);
void InvertAxisForBinding(struct FName Action, TArray<struct FPlayerBinding>& OutBindings);
void InvertAxis(struct FName Axis);
void RemoveBinding(struct FName Action, struct FName Key, uint8_t AxisSign, TArray<struct FPlayerBinding>& OutBindings);
void SetSingleBinding(struct FName Action, struct FName Key, uint8_t AxisSign, TArray<struct FPlayerBinding>& DefaultBindings, TArray<struct FPlayerBinding>& OutBindings);
void SetBinding(struct FName Action, struct FName Key, uint8_t AxisSign, TArray<struct FPlayerBinding>& DefaultBindings, TArray<struct FPlayerBinding>& OutBindings);
void SetCurrentBinding(struct FName Key, unsigned long bGamepad, uint8_t AxisSign);
void HandleOverwriteCustomBindingPreset(struct FName Key, unsigned long bGamepad, uint8_t AxisSign);
bool HandleBindingInputAxis(int32_t ControllerId, struct FName Key, float delta, float DeltaTime,
unsigned long bGamepad);
bool HandleBindingInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float
AmountDepressed, unsigned long bGamepad);
void ClearBindingFromArray(struct FName Action, TArray<struct FPlayerBinding>&
DefaultBindings, TArray<struct FPlayerBinding>& OutBindings);
void ClearKeyBinding(struct FName Action);
void HandleOverwriteCustomBindingsClear(struct FName Action);
void ClearBinding(struct FName Action);
void CancelRecordBinding();
void StartRecordBinding(struct FName Action);
class FString LocalizeKeyInternal(struct FName Key);
class FString LocalizeKey(struct FName Key, uint8_t AxisSign);
struct FLocalizedAction GetLocalizedAction(struct FName ActionName);
void PopulateSettingsMenu();
void BuildKeyBindings();
void HandleProfileSet(class ULocalPlayer_TA* InLocalPlayer);
void eventOnShellSet();
};

// Class TAGame.__GFxData_Controls_TA_SetCurrentBinding_0x1
// 0x000D (0x0060 - 0x006D)
class U_GFxDATA_Controls_TA_SetCurrentBinding_0x1 : public UObject
{
public:
    struct FName           Key;          // 0x0060 (0x0008)
    [0x0000000000000000] 
    unsigned long          bGamepad : 1; // 0x0068 (0x0004)
    [0x0000000000000000] [0x00000001]
    uint8_t                AxisSign;    // 0x006C (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Controls_TA_SetCurrentBinding_0x1");
}

return uClassPointer;
};

void __GFxData_Controls_TA_SetCurrentBinding_0x1(class UGFxModal_X* _);
};

// Class TAGame._GFxData_DLC_TA_HandleEntitlementsError_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_DLC_TA_HandleEntitlementsError_0x1 : public UObject
{
public:
class UError* Error; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_DLC_TA_HandleEntitlementsError_0x1");
}

return uClassPointer;
};

void __GFxData_DLC_TA_HandleEntitlementsError_0x1(class UGFxModal_X* Modal);
};

// Class TAGame.GFxData_DLC_TA
// 0x0038 (0x0098 - 0x00D0)
class UGFxData_DLC_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGfxLCDData> Packs; // 0x0098 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bReturningFromStore : 1; // 0x00A8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
class UMtxConfig_TA* MtxConfig; // 0x00B0 (0x0008)
[0x0000800000000001] (CPF_Edit)
class UOnlineSubsystem* OnlineSub; // 0x00B8 (0x0008)
[0x0000800000000001] (CPF_Edit)
class UOnlinePlayerMTX_TA* MTX; // 0x00C0 (0x0008)
}

```

```

[0x0000000000000000]
uint64_t TimeWhenClientLostFocus; // 0x00C8 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_DLC_TA");
}

return uClassPointer;
};

struct FProductInstanceID __GFxData_DLC_TA__HandleClaimEntitlements_0x1(class UOnlineProduct_TA* P);
class UShowroomMetrics_TA* GetShowroomMetrics();
void OnClaimedEntitlements(TArray<struct FProductInstanceID> InstanceID);
void HandleClaimEntitlements(class UOnlinePlayerMTX_TA* InMTX, TArray<class UOnlineProduct_TA*> NewProducts);
void HandleEntitlementsError(class UOnlinePlayerMTX_TA* InMTX, class UError* Error);
void OnExternalUIChange(unsigned long bIsOpening);
void CheckNewPurchases();
void HandleUnlockedDLCChanged();
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.MtxConfig_TA
// 0x0050 (0x0078 - 0x00C8)
class UMtxConfig_TA : public UOnlineConfig_X
{
public:
unsigned long bAllowMicroTransactions : 1; // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bAllowCurrencyTrading : 1; // 0x0078 (0x0004)
[0x0000000000004001] [0x00000002] (CPF_Edit | CPF_Config)
unsigned long bAllowCurrencyGifting : 1; // 0x0078 (0x0004)
[0x0000000000004001] [0x00000004] (CPF_Edit | CPF_Config)
unsigned long bAllowCurrencyForCurrencyTrades : 1; // 0x0078
(0x0004) [0x0000000000004001] [0x00000008] (CPF_Edit | CPF_Config)
unsigned long bEnablePlatformStartPurchase : 1; // 0x0078
(0x0004) [0x0000000000000001] [0x00000010] (CPF_Edit)
int32_t MaxCurrencyWalletAmount; // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t MaxCurrencyDailyAmount; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
class FString FulfillmentRestrictedWalletMessage; // 0x0088 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString FulfillmentRestrictedDailyMessage; // 0x0098 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

```

```
class FString           FulfillmentRestrictedTitle;          // 0x00A8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           CurrencyDelimiter;                // 0x00B8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MtxConfig_TA");
}

return uClassPointer;
};

bool CreateModalForError(class UGFxShell_TA* Shell, class UError* Error);
void InitModalForError(class UGFxModal_X* Modal, class UErrorType* Error);
void InitErrorModalWithBody(class UGFxModal_X* Modal, class FString BodyString);
class FString LocalizeFulfillmentRestrictedMessage(class FString Message);
};

// Class TAGame.__GFxDATA_EOSVoiceManager_TA__HandleFriendsListChanged_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxDATA_EOSVoiceManager_TA__HandleFriendsListChanged_0x1 : public UObject
{
public:
class FString           CurrentRoomName;              // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_EOSVoiceManager_TA__HandleFriendsListChanged_0x1");
}

return uClassPointer;
};

bool __GFxDATA_EOSVoiceManager_TA__HandleFriendsListChanged_0x1(class
UGFxData_EOSVoiceRoom_TA* P);
};

// Class TAGame.GFxData_EOSVoiceRoom_TA
// 0x014C (0x0094 - 0x01E0)
class UGFxDATA_EOSVoiceRoom_TA : public UGFxDataRow_X
{
```

```

public:
class FString RoomName; // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint8_t RoomType; // 0x00A8 (0x0001)
[0x0001000040000000] (CPF_EditInlineNotify)
class FString RoomTitle; // 0x00B0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString PartyRoomTitle; // 0x00C0 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString MatchRoomTitle; // 0x00D0 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString SettingDisabledTitle; // 0x00E0 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString SettingDisabledDesc; // 0x00F0 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString MatchTypeTitle; // 0x0100 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString PartyTypeTitle; // 0x0110 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString SpectatorRoomTitle; // 0x0120 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString SpectatorTypeTitle; // 0x0130 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString RemotePlayerJoinedRoom; // 0x0140 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString PlayerJoinedRoom; // 0x0150 (0x0010)
[0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString RoomTypeString; // 0x0160 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
TArray<class UGFxData_EOSVoiceRoomMember_TA*> RoomMembers; //
0x0170 (0x0010) [0x000100000400000] (CPF_NeedCtorLink)
class UEOSVoiceSettingsSave_TA* VoiceSettingsSave; // 0x0180
(0x0008) [0x000100000002000] (CPF_Transient)
unsigned long bManuallyJoined : 1; // 0x0188 (0x0004)
[0x000100000000000] [0x00000001]
TArray<struct FNotificationCooldown> NotificationCooldowns; // 0x0190
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)
uint64_t CreatedTimestamp; // 0x01A0 (0x0008)
[0x000100000000000]
struct FVoiceRoomPlayerActivity PlayerActivity; // 0x01A8 (0x0020)
[0x000100000400000] (CPF_NeedCtorLink)
class UEOSVoiceManager_TA* VoiceManager; // 0x01C8
(0x0008) [0x0001800000000000]
class UPersonas_TA* Personas; // 0x01D0 (0x0008)
[0x000180000000000]
class UEOSVoiceConfig_TA* VoiceConfig; // 0x01D8 (0x0008)
[0x000180000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSVoiceRoom_TA");
}

return uClassPointer;
};

void __GFxData_EOSVoiceRoom_TA__Init_0x2(class UEOSVoiceSettingsSave_TA* VoiceSave);
void __GFxData_EOSVoiceRoom_TA__Init_0x1(struct FMappedRoomMember M);
void __GFxData_EOSVoiceRoom_TA__HandleLeftVoiceRoom_0x2(class
UGFxData_EOSVoiceRoomMember_TA* P);
bool __GFxData_EOSVoiceRoom_TA__HandleLeftVoiceRoom_0x1(class
UGFxData_EOSVoiceRoomMember_TA* P);
void __GFxData_EOSVoiceRoom_TA__HandleSplitscreenPlayerLoggedIn_0x1();
bool __GFxData_EOSVoiceRoom_TA__RecordVoiceRoomActivity_0x1(class
UGFxData_EOSVoiceRoomMember_TA* P);
void __GFxData_EOSVoiceRoom_TA__GetPlayerVoiceActivity_0x1(struct FActivePlayerVoice P);
void eventOnRemoved();
bool PlayerOwnsThisShell(class UOnlinePlayer_X* OnlinePlayer);
class UGFxData_EOSVoiceRoomMember_TA* GetMemberByPlayerId(struct FUniqueNetId
PlayerID);
class UGFxData_EOSVoiceRoomMember_TA* GetMemberByEpicId(class FString EpicAccountId);
void GetPlayerVoiceActivity(class UGFxData_EOSVoiceRoomMember_TA* Member);
void RecordVoiceRoomActivity();
void HandlePlayerStatusChanged(class FString EpicAccountId, class FString InRoomName,
struct FVoiceRoomMemberStatus MemberStatus);
void SendChatNotification(class FString EpicAccountId, class FString BaseMessage);
void HandleRemotePlayerLeftRoom(class FString EpicAccountId, class FString InRoomName);
bool CanSendChatNotification(class FString EpicAccountId);
void OnVoiceError(class UError* VoiceError);
void CheckPlayerPermissionError(class UErrorType* ErrorTypeOverride);
bool ShellOwnerIsInRoom();
void HandleRemotePlayerJoinedRoom(struct FMappedRoomMember RoomMember, class
FString InRoomName);
void HandleSplitscreenPlayerLoggedIn(class UOnlinePlayer_X* OnlinePlayer);
void HandleSplitscreenPlayerJoinedSession(class ULocalPlayer* NewPlayer);
void HandleLeftVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString InRoomName,
unsigned long bSuccess);
void CheckPlayerStatusOnJoin(class UGFxData_EOSVoiceRoomMember_TA* Member);
void HandleJoinedVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString InRoomName,
unsigned long bSuccess);
void RemoveRoomMember(class UGFxData_EOSVoiceRoomMember_TA* Member, unsigned
long bAlertRemoval);
void RemoveRoomMemberById(class FString EpicAccountId);
void HandleBlockStatusChanged(class UPersona_TA* Persona);
void HandleMuteStatusChanged(class UGFxData_EOSVoiceRoomMember_TA* Member,
unsigned long bMuted);
void OnUserChatPermissionChanged(struct FUniqueNetId PlayerID, struct FUniqueNetId
TargetId, unsigned long bAllowed);
void CreateRoomMember(struct FMappedRoomMember RoomMember);
void HandlePlayerVoiceError(class UOnlinePlayer_X* OnlinePlayer, class UError* VoiceError);
void JoinRoom();
void LeaveRoom();
void Init(struct FVoiceRoom VoiceRoomData);

```

```

};

// Class TAGame.GFxData_EOSVoiceManager_TA
// 0x0050 (0x0098 - 0x00E8)
class UGFxData_EOSVoiceManager_TA : public UGFxDataSingleton_X
{
public:
TArray<class UGFxData_EOSVoiceRoom_TA*> TrackedRooms; // 0x0098 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class UEOSVoiceSettingsSave_TA* VoiceSettingsSave; // 0x00A8 (0x0008) [0x0001000000002000] (CPF_Transient)
class USoundSettingsSave_TA* SoundSettingsSave; // 0x00B0 (0x0008) [0x0001000000002000] (CPF_Transient)
class UUISavedValues_TA* UISavedValues; // 0x00B8 (0x0008) [0x0001000000002000] (CPF_Transient)
unsigned long bPushToTalkActive : 1; // 0x00C0 (0x0004) [0x0001000000000000] [0x00000001]
unsigned long bInMapTransition : 1; // 0x00C0 (0x0004) [0x0001000000000000] [0x00000002]
unsigned long bMainMenuNotifications : 1; // 0x00C0 (0x0004) [0x0001000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long bMatchNotifications : 1; // 0x00C0 (0x0004) [0x0001000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bVoiceSettingEnabled : 1; // 0x00C0 (0x0004) [0x0001000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long bDisplayVoiceTab : 1; // 0x00C0 (0x0004) [0x0001000040000000] [0x00000020] (CPF_EditInlineNotify)
unsigned long bVoiceFeatureEnabled : 1; // 0x00C0 (0x0004) [0x0001000040000000] [0x00000040] (CPF_EditInlineNotify)
float InactivePTTVolume; // 0x00C4 (0x0004) [0x0001000000000002] (CPF_Const)
float ActivePTTVolume; // 0x00C8 (0x0004) [0x0001000000000002] (CPF_Const)
class UGameViewportClient_X* ViewportClient; // 0x00D0 (0x0008) [0x0001000000002000] (CPF_Transient)
class UEOSVoiceManager_TA* VoiceManager; // 0x00D8 (0x0008) [0x0008] [0x0001800000000000]
class UEOSVoiceConfig_TA* VoiceConfig; // 0x00E0 (0x0008) [0x0001800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSVoiceManager_TA");
}

return uClassPointer;
}

void __GFxData_EOSVoiceManager_TA__OnRemoved_0x1(class UGFxData_EOSVoiceRoom_TA*

```

```
P);
void __GFxData_EOSVoiceManager_TA__HandlePsyNetConnected_0x1(class
UUISavedValues_TA* SaveObject);
bool __GFxData_EOSVoiceManager_TA__HandleVoiceChatSettingChanged_0x2(class
UGFxData_EOSVoiceRoom_TA* P);
static void HandleUserChatPermissionsChanged(class UGFxData_EOSVoiceManager_TA*
GFxVoiceManager, class UTargetUserChatPermChangedEvent* ChangedEvent);
static void HandleAudioDevicesUpdated(class UGFxData_EOSVoiceManager_TA*
GFxVoiceManager, class UAudioDevicesChangedEvent* _);
void HandlePostLoadMap();
void HandlePreLoadMap(class FString MapName);
bool ShouldFocusMute();
float GetOutputVolume();
float GetInputVolume();
void UpdatePlayerAudioVolume();
bool HandleInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float
AmountDepressed, unsigned long bGamepad);
void HandleFocusChanged(unsigned long bHasFocus);
void UpdatePlayersAudioDevices();
bool IsTerminalError(class UError* InError);
bool CanViewVoiceChat();
void HandleLeftVoiceRoom(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName,
unsigned long bSuccess);
void OnVoiceDataRemoved();
void HandleTrackedVoiceRoomChanged(class UOnlinePlayer_X* OnlinePlayer, struct
FVoiceRoom NewVoiceRoom);
bool SetAudioOutput(float Volume);
bool SetAudioInput(float Volume);
void HandleOutputAudioDeviceChanged();
void HandleInputAudioDeviceChanged();
void HandleNotificationsChanged();
void HandleVoiceInputModeChanged();
void HandleVoiceChatSettingChanged();
void HandleFriendsListChanged(class UOnlinePlayerFriends_X* FriendsRef, class UError*
InError);
void HandleSoundSettingsUnloaded(class USoundSettingsSave_TA* InSoundSave);
void HandleSoundSettingsLoaded(class USoundSettingsSave_TA* InSoundSave);
void HandleVoiceSettingsUnloaded(class UEOSVoiceSettingsSave_TA* InVoiceSettingsSave);
void HandleVoiceSettingsLoaded(class UEOSVoiceSettingsSave_TA* InVoiceSettingsSave);
void UpdateFeatureEnabled();
void HandleVoiceConfig();
void HandleSaveSyncSuccess(class UOnlineStorageSyncManager_TA*
OnlineStorageSyncManager);
void HandleSaveLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*
SaveData, class UError* Error);
void HandlePsyNetConnected(class UPsyNetConnection_X* Connection);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.__GFxData_EOSVoiceManager_TA__HandleUserChatPermissionsChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_EOSVoiceManager_TA__HandleUserChatPermissionsChanged_0x1 : public
UObject
```

```
{  
public:  
class UTargetUserChatPermChangedEvent* ChangedEvent; // 0x0060  
(0x0008) [0x0001000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__GFxDATA_EOSVoiceManager_TA__HandleUserChatPermissionsChanged_0x1");  
}  
  
return uClassPointer;  
};  
  
void __GFxDATA_EOSVoiceManager_TA__HandleUserChatPermissionsChanged_0x1(class  
UGFxDATA_EOSVoiceRoom_TA* P);  
};  
  
// Class TAGame.__GFxDATA_EOSVoiceManager_TA__HandleVoiceChatSettingChanged_0x1  
// 0x0010 (0x0060 - 0x0070)  
class U_GFxDATA_EOSVoiceManager_TA__HandleVoiceChatSettingChanged_0x1 : public  
UObject  
{  
public:  
class FString CurrentRoomName; // 0x0060 (0x0010)  
[0x0001000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__GFxDATA_EOSVoiceManager_TA__HandleVoiceChatSettingChanged_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __GFxDATA_EOSVoiceManager_TA__HandleVoiceChatSettingChanged_0x1(class  
UGFxDATA_EOSVoiceRoom_TA* P);  
};  
  
// Class TAGame.__GFxDATA_EOSVoiceRoom_TA__CanSendChatNotification_0x1  
// 0x0010 (0x0060 - 0x0070)  
class U_GFxDATA_EOSVoiceRoom_TA__CanSendChatNotification_0x1 : public UObject  
{
```

```
public:
class FString EpicAccountId; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_EOSVoiceRoom_TA__CanSendChatNotification_0x1");
}

return uClassPointer;
};

bool __GFxDATA_EOSVoiceRoom_TA__CanSendChatNotification_0x1(struct
FNotificationCooldown P);
};

// Class TAGame.__GFxDATA_EOSVoiceRoom_TA__GetMemberByEpicId_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxDATA_EOSVoiceRoom_TA__GetMemberByEpicId_0x1 : public UObject
{
public:
class FString EpicAccountId; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_EOSVoiceRoom_TA__GetMemberByEpicId_0x1");
}

return uClassPointer;
};

bool __GFxDATA_EOSVoiceRoom_TA__GetMemberByEpicId_0x1(class
UGFxDATA_EOSVoiceRoomMember_TA* M);
};

// Class TAGame.GFxDATA_EOSVoiceRoomMember_TA
// 0x00A4 (0x0094 - 0x0138)
class UGfxDATA_EOSVoiceRoomMember_TA : public UGfxFDataRow_X
{
public:
class FString EpicAccountId; // 0x0098 (0x0010)
```

```

[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FUniqueNetId           PlayerID;          // 0x00A8 (0x0048)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString                 RoomName;         // 0x00F0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long                 bLocal : 1;        // 0x0100 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long                 bSpeaking : 1;      // 0x0100 (0x0004)
[0x0001000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long                 bSelfMuted : 1;     // 0x0100 (0x0004)
[0x0001000040000000] [0x00000004] (CPF>EditInlineNotify)
unsigned long                 bLocalMuted : 1;    // 0x0100 (0x0004)
[0x0001000040000000] [0x00000008] (CPF>EditInlineNotify)
unsigned long                 bAdminMuted : 1;    // 0x0100 (0x0004)
[0x0001000040000000] [0x00000010] (CPF>EditInlineNotify)
unsigned long                 bBlocked : 1;       // 0x0100 (0x0004)
[0x0001000040000000] [0x00000020] (CPF>EditInlineNotify)
class UPersona_TA*           Persona;          // 0x0108 (0x0008)
[0x000100000002000] (CPF_Transient)
TArray<struct FActivePlayerVoice> PlayerActivity; // 0x0110 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventMuteStatusChanged__Delegate; // 0x0120
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSVoiceRoomMember_TA");
}

return uClassPointer;
};

void __GFxData_EOSVoiceRoomMember_TA__Init_0x4(class UObject* _);
void __GFxData_EOSVoiceRoomMember_TA__Init_0x3(class UObject* _);
void __GFxData_EOSVoiceRoomMember_TA__Init_0x2(class UObject* _);
void __GFxData_EOSVoiceRoomMember_TA__Init_0x1(class AGameEvent_TA* _);
void ResetStatus();
bool TrySetPlayerMuteStatus(unsigned long bMuted);
void SetPlayerMuteStatus(unsigned long bMuted);
void MarkLastPlayerActivityEnded();
void AddNewPlayerActivityEntry();
void UpdatePlayerStatus(struct FVoiceRoomMemberStatus& MemberStatus);
void FinalizeActivityData();
uint8_t GetPlayerLocation();
uint64_t GetEpochTime();
void HandleCurrentGameChanged();
bool IsShellOwner();
void Init(struct FMappedRoomMember RoomMember, class FString InRoomName, class
UPersona_TA* InPersona);

```

```
void EventMuteStatusChanged(class UGFxData_EOSVoiceRoomMember_TA* GFxRow, unsigned long bMuted);
};

// Class TAGame.__GFxDATA_EOSVoiceRoom_TA__GetMemberByPlayerId_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxDATA_EOSVoiceRoom_TA__GetMemberByPlayerId_0x1 : public UObject
{
public:
    struct FUniqueNetId           PlayerID;                      // 0x0060 (0x0048)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_EOSVoiceRoom_TA__GetMemberByPlayerId_0x1");
        }

        return uClassPointer;
    }

    bool __GFxDATA_EOSVoiceRoom_TA__GetMemberByPlayerId_0x1(class
UGFxData_EOSVoiceRoomMember_TA* M);
};

// Class TAGame.__GFxDATA_EOSVoiceRoom_TA__HandleBlockStatusChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_EOSVoiceRoom_TA__HandleBlockStatusChanged_0x1 : public UObject
{
public:
    class UPersona_TA*           Persona;                      // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_EOSVoiceRoom_TA__HandleBlockStatusChanged_0x1");
        }

        return uClassPointer;
    }

    bool __GFxDATA_EOSVoiceRoom_TA__HandleBlockStatusChanged_0x1(class
UGFxData_EOSVoiceRoomMember_TA* M);
```

```

};

// Class TAGame.__GFxData_EpicLogin_TA__RequestAccountAuthorization_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_EpicLogin_TA__RequestAccountAuthorization_0x1 : public UObject
{
public:
class UEpicLogin_X* EpicLogin; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_EpicLogin_TA__RequestAccountAuthorization_0x1");
}

return uClassPointer;
};

void __GFxDATA_EpicLogin_TA__RequestAccountAuthorization_0x1(class FString AuthTicket,
class FString __, class UError* Error);
};

// Class TAGame.GFxData_EpicLogin_TA
// 0x00D8 (0x0098 - 0x0170)
class UGFxDATA_EpicLogin_TA : public UGFxDATASingleton_X
{
public:
class UEpicAccountSave_TA* EpicAccountSave; // 0x0098
(0x0008) [0x0000000000002000] (CPF_Transient)
class FString EpicDisplayName; // 0x00A0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString AccountLinkURL; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bEnforceCabinedMode : 1; // 0x00C0 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bInCabinedMode : 1; // 0x00C0 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bConnectionLost : 1; // 0x00C0 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
class FString SetDOBErrorMessage; // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString SetParentEmailErrorMessage; // 0x00D8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ConfirmDisplayNameErrorMessage; // 0x00E8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString UnsupportedCorrectiveActionURL; // 0x00F8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString EpicPlayerName; // 0x0108 (0x0010)

```

```
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ParentalControlsURL; // 0x0118 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ActivateURL; // 0x0128 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString AccountURL; // 0x0138 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString TOSURL; // 0x0148 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString PrivacyPolicyURL; // 0x0158 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UEpicConfig_X* EpicConfig; // 0x0168 (0x0008)
[0x0000804000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EpicLogin_TA");
}

return uClassPointer;
};

void __GFxData_EpicLogin_TA__OnShellSet_0x1(class UEpicAccountSave_TA*
EpicAccountSaveData);
bool __GFxData_EpicLogin_TA__HandleConfigChanged_0x1(class ULocalizedAccountLinkURL*
A);
void LostConnection();
bool WillShowCabinedReminder();
bool IsAnyLocalPlayerCabinedOrGuest();
void UnsupportedCorrectiveActionVerify();
void EpicDisplayNameScreenDone();
bool AskConfirmDisplayName();
bool IsPrimaryPlayerXboxGuest();
bool AskParentEmail();
bool AskAge();
void SetParentEmail(class FString Email);
bool ValidateEmail(class FString Email);
bool ValidateDOB(int32_t Year, int32_t Month, int32_t Day);
void SetDOB(int32_t Year, int32_t Month, int32_t Day);
void ReceivedAccountAuthorization(unsigned long bSuccess);
void RequestAccountAuthorization();
void OnCompletedLinkingFlow();
void OnLoadedEpicAccountSaveData(unsigned long bPlatformLinked);
void HandleSaveSyncSuccess(class UOnlineStorageSyncManager_TA*
OnlineStorageSyncManager);
void HandleSaveLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*
SaveData, class UError* Error);
void CloseAgeGate();
void ShowCabinedReminder();
```

```

void ShowXboxGuestIsPrimaryPlayerWarningScreen();
void ShowWelcomeNotCabinModeScreen();
void ShowUnsupportedCorrectiveActionScreen();
void ShowWelcomeToCabinModeScreen();
void ConfirmDisplayNameReturned();
void ShowEpicDisplayNameScreen();
void SetParentEmailReturned();
void ShowParentEmailScreen();
void SetDOBReturned();
void ShowDOBScreen();
void ParentalConsentVerification();
void ShowParentalConsentScreen();
void ShowChoosePrimaryScreen();
void ShowEpicAccountLinkScreen();
void AccountPortalCompletedAttemptReLogin();
void LaunchAccountPortalURL();
void RequestPinGrantCode();
void PinGrantRequestFailed(class FString Error);
void NotifyOneAccountModalShown();
void OnSkippedChoosePrimaryScreen();
void OnSkippedEpicAccountLinkScreen();
void CheckForLoginErrors();
void HandleFailedConnectionModalDismissed(class UGFxModal_X* Modal);
void HandleLoggedIn();
void HandleLoginError(class UEpicLogin_X* EpicLogin, class UErrorType* Error);
void HandleConfigChanged();
class UEpicLogin_X* GetEpicLogin();
void eventOnShellSet();
};

// Class TAGame.__GFxDATA_ErrorModals_TA__AlertError_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_ErrorModals_TA__AlertError_0x1 : public UObject
{
public:
    class UError*           Error;           // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_ErrorModals_TA__AlertError_0x1");
        }

        return uClassPointer;
    }

    bool __GFxDATA_ErrorModals_TA__AlertError_0x1(struct FPendingError PE);
};

```

```

// Class TAGame.GFxData_ErrorModals_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_ErrorModals_TA : public UGFxDataSingleton_X
{
public:
    class AGameEvent_TA*           GameEvent;          // 0x0098 (0x0008)
    [0x0000800000000001] (CPF_Edit)
    TArray<struct FPendingError>   PendingErrors;     // 0x00A0 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate         _OnErrorModalCreated__Delegate; // 0x00B0
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_ErrorModals_TA");
        }

        return uClassPointer;
    }

void __GFxData_ErrorModals_TA__OnShellSet_0x4(struct FName PrevMenu, struct FName
TopMenu, unsigned long bClearingStack);
void __GFxData_ErrorModals_TA__OnShellSet_0x3(class AGameInfo_GFxMenu_TA* GameInfo,
uint8_t UIState);
void __GFxData_ErrorModals_TA__OnShellSet_0x2(class AGameEvent_TA*_);
void __GFxData_ErrorModals_TA__OnShellSet_0x1(class AGameInfo_X*_);
void __GFxData_ErrorModals_TA__ShowError_0x1(class UGFxModal_X*_);
void ShowError(unsigned long bForce);
bool CanShowError();
void RemoveErrorType(class UErrorType* InErrorType);
void AlertError(class UError* Error, struct FScriptDelegate Callback, class UClass* InModalClass);
void eventOnShellSet();
void OnErrorModalCreated(class UGFxModal_X* Modal);
};

// Class TAGame.__GFxData_ErrorModals_TA__RemoveErrorType_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_ErrorModals_TA__RemoveErrorType_0x1 : public UObject
{
public:
    class UErrorType*             InErrorType;        // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ErrorModals_TA__RemoveErrorType_0x1");
}

return uClassPointer;
};

bool __GFxData_ErrorModals_TA__RemoveErrorType_0x1(struct FPendingError P);
};

// Class TAGame.__GFxData_Facelt_TA__HandleError_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Facelt_TA__HandleError_0x1 : public UObject
{
public:
class UError* InError; // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Facelt_TA__HandleError_0x1");
}

return uClassPointer;
};

void __GFxData_Facelt_TA__HandleError_0x1();
};

// Class TAGame.FaceltErrors_TA
// 0x0058 (0x0080 - 0x00D8)
class UFaceltErrors_TA : public UErrorList
{
public:
class UErrorType* MatchHasBeenCancelled; // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* CannotJoinCrossPlatformMatch; // 0x0088
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* CannotJoinWhileInParty; // 0x0090 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* CannotJoinWhilePlayingSplitscreen; // 0x0098
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* CannotJoinWhileOnline; // 0x00A0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* ReturnToMainMenuToJoin; // 0x00A8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* JoinGameFailed; // 0x00B0 (0x0008)

```

```

[0x0001000000000002] (CPF_Const)
class UErrorType*           NoFaceltMatchFound;          // 0x00B8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           FaceltGameServerPending;    // 0x00C0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           ExternalMatchServerNotFound; // 0x00C8
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           ReservationSentToIncorrectPlayer; // 0x00D0
(0x0008) [0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FaceltErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Facelt_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_Facelt_TA : public UGFxDataSingleton_X
{
public:
unsigned long             bEnabled : 1;                // 0x0098 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
class UFacelt_TA*          Facelt;                  // 0x00A0 (0x0008)
[0x0001800000002001] (CPF>Edit | CPF_Transient)
class UFaceltConfig_TA*     Config;                  // 0x00A8 (0x0008)
[0x0001800000002000] (CPF_Transient)
class UPsyNet_X*            PsyNet;                 // 0x00B0 (0x0008)
[0x0001800000000000]
uint64_t                   LastSpectatorPopupTime; // 0x00B8 (0x0008)
[0x0001000000000000]
uint64_t                   SpectatorPopupDelaySeconds; // 0x00C0 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Facelt_TA");
}

return uClassPointer;
}

```

```

};

class FString GetPartnerString(int32_t PartnerID);
void HandleFaceltMatchFound(class UPsyNetService_FaceltSearchStarted_TA* Service);
void HandleJoiningReservation(class UAsyncTask* Task, struct FFaceltMatchInfo& MatchInfo);
void HandleAddReservationCommon(uint64_t Expiration, unsigned long bSuppressNotification,
struct FServerReservationData& ReservationData, struct FFaceltMatchInfo& MatchInfo);
void HandleClienFaceltClientReservationMessage(class UIReservationConnection_X*
Connection, class UFaceltClientReservationMessage_X* Message);
void FindMatchForPlayer();
void HandlePsyNetLogin();
void HandleFaceltConfigChange();
void HandleError(class UError* InError);
void HandleFaceltRefChange();
void eventOnShellSet();
};

// Class TAGame.GFxShell_TA
// 0x0170 (0x0130 - 0x02A0)
class UGFxShell_TA : public UGFxShell_X
{
public:
    class UChampionshipTrophy_TA*          ChampionshipTrophyArchetype;      // 0x0130 (0x0008) [0x0000000000000001] (CPF_Edit)
    class ULocalPlayer_TA*                  TAPlayer;                      // 0x0138 (0x0008) [0x0000004000002000] (CPF_Transient)
    class UGFxData_System_TA*              SystemData;                     // 0x0140 (0x0008) [0x0000004000002000] (CPF_Transient)
    class UGFxData_LocalPlayer_TA*         LocalPlayerData;                // 0x0148 (0x0008) [0x0000004000002000] (CPF_Transient)
    unsigned long                         bEnableFocusDebugger : 1;        // 0x0150 (0x0004) [0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
    unsigned long                         bDisableStartupMovies : 1;       // 0x0150 (0x0004) [0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
    unsigned long                         bAllowOnlineStorageResyncModal : 1; // 0x0150 (0x0004) [0x0000000000004002] [0x00000004] (CPF_Const | CPF_Config)
    class FString                          EOSReconnectMessage;           // 0x0158 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          EOSNotConnectedMessage;        // 0x0168 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          EOSRestartLauncherMessage;     // 0x0178 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          SaveReplayMessage;             // 0x0188 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          DefaultDisconnectionTitle;     // 0x0198 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          PsyNetReconnectMessage;        // 0x01A8 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          PSPlusTipTitle;                // 0x01B8 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          PSPlusTipMessage;              // 0x01C8 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                          RankedReconnectTitle;          // 0x01D8 (0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
};

```

```

class FString           RankedReconnectBody;          // 0x01E8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           BannedTimeRemainingMessage;    // 0x01F8
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           PermaBannedMessage;          // 0x0208 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
TArray<class UGFxModal_X*>      ActiveModals;          // 0x0218 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
uint8_t                 PendingMessageType;          // 0x0228 (0x0001)
[0x0000000000002000] (CPF_Transient)
class FString           PendingErrorTitle;          // 0x0230 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString           PendingErrorMessage;         // 0x0240 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
float                  LastPsyNetReconnectTime;     // 0x0250 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UGFxModal_X*      ModalKick;                // 0x0258 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString           BuildEdition;              // 0x0260 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FScriptDelegate  __EventModalOpened__Delegate; // 0x0270
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate  __EventModalClosed__Delegate; // 0x0288
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxShell_TA");
}

return uClassPointer;
};

void __GFxShell_TA__HandleOnlinePlayerStorageSyncError_0x1(class UGFxModal_X* _);
void __GFxShell_TA__HandleSaveDataLoaded_0x1(class UOnlineStorageSyncManager_TA* _,
class UAsyncTask* Task);
static bool __GFxShell_TA__InitPlaylistSkills_0x1(class UGameSettingPlaylist_X* P);
static class FString ReplaceHtmlEntities(class FString Text);
static class UGFxDataStore_X* GetData(class ULocalPlayer_X* ForPlayer);
static class UGFxShell_TA* GetShell(class ULocalPlayer_X* ForPlayer);
void HandleTopMenuChange(struct FName PrevMenu, struct FName TopMenu, unsigned long bClearingStack);
static class UGFxModal_X* InitErrorModal(class UGFxModal_X* Modal, class FString Error, class FString Title);
class UGFxModal_X* AlertErrorObj(class UError* Error);
class UGFxModal_X* AlertError(class FString Error, class FString Title, class UClass* ModalClass);
bool AnyPlayerChatRestricted();
class FString GetPaymentServices();

```

```
class FString GetSCT();
class FString GetPrivacyPolicy();
class FString GetTermsOfService();
class FString GetEULA();
void OnRankedReconnect(struct FServerReservationData Reservation);
void ShowRankedReconnectModal(struct FServerReservationData Reservation);
void OnGotoStorePage(class UGFxModal_X* Modal);
void ShowUnauthorizedModal();
static void InitPlaylistSkills(class UGFxShell_TA* InShell, class UDownloadedPlaylistsData_X*
DownloadedPlaylists, class USyncedSkillData_X* Skill);
void OnPlaylistsUpdated();
void SetInputActionEnabled(struct FName ActionName, unsigned long bEnabled);
void VoteToForfeit();
class UGFxModal_X* ShowTextInput(class FString Title, class FString Body, struct
FScriptDelegate Callback, unsigned long bDisplayAsPassword, class FString DefaultText, int32_t
MaxLength);
void HandleCommerceDialogClosed();
void SetViewingOnlineScene(unsigned long bViewing);
void TryReconnectingToEpicOnlineServices(class UOnlinePlayer_X* OnlinePlayer);
void HandlePsyNetLoginAttempt(class UOnlinePlayerAuthentication_X* Auth);
void HandlePsyNetLoginConfirm(class UGFxModal_X* Modal);
class FString GetPrivilegeRestrictionMessage(class UError* FailReason);
void HandleCrewUnknownState(class UPsyNetService_CrewUnknownStatus_TA* Service);
void HandlePrivilegeCheckRestriction(class UPrivilegeCheck_X* PrivilegeCheck);
void HandleOnlineGameConnectionError(class UError* Error, class UOnlinePlayer_X*
OnlinePlayer);
void HandlePsyNetDisabled(class UPsyNetConnection_X* Connection);
void HandleBanned(class UOnlinePlayerAuthentication_X* Auth);
void HandlePauseChanged();
void PopUISoundState(struct FName StateName);
void PushUISoundState(struct FName StateName);
void CheckoutCompleteFlashNNX();
void OpenStoreOverlay();
void OnKickPlayerAccepted(class UGFxModal_X* Modal);
void OnKickPlayerModalClosed(class UGFxModal_X* Modal);
void HandleCloseKickPlayerDialog();
void HandleDrawWorldFullScreenChanged(class UGameViewportClient_TA* GVC);
void HandleSplitScreenTypeChanged(class UGameViewportClient_TA* GVC);
void ShowControllerPickerUI();
void ShowControllerUI();
void ShowKickPlayerModal(int32_t ControllerIndex);
void ShowLoginUI(int32_t ControllerIndex);
void HandleKickedResponse(class UGFxModal_X* Modal);
void HandleUserSaveReplay(class UGFxModal_X* Modal);
void ShowErrorMessage();
void HandleGameInit(class AGRI_X* GRI);
void HandleConnectionError(uint8_t MessageType, class FString Title, class FString Message);
void OnCancelSaveSaveData(class UGFxModal_X* Modal);
void OnRetrySaveSaveData(class UGFxModal_X* Modal);
class UGFxModal_X* CreateSaveErrorModal(class UError* Error);
void HandleSaveDataSaved(class USaveGameManager_TA* SaveManager, class USaveData_TA*
SaveData, class UError* Error);
void NotifyEnableSaveDataProcess(class UGFxModal_X* Modal);
void OnDisableSaveData(class UGFxModal_X* Modal);
```

```
void OnLoadFailureDisableSaveData(class UGFxModal_X* Modal);
void OnCreateNewSaveData(class UGFxModal_X* Modal);
void OnConfirmCreateNewSaveData(class UGFxModal_X* Modal);
void OnRetryLoadSaveData(class UGFxModal_X* Modal);
void HandleSaveDataLoaded(class USaveGameManager_TA* SaveManager, class
USaveData_TA* SaveData, class UError* Error);
void HandleOnlinePlayerStorageSyncError(class UError* Error);
void DelayedNotifyWhenSaveDataLoaded();
void HandleModalClosed(class UGFxModal_X* Modal);
class UGFxObject* OnCreateCinematicModal();
class UGFxObject* OnCreateTutorialModal();
class UGFxObject* OnCreateModal(class FString ClassType);
void OnShowModal(class UGFxObject* Modal);
class UGFxModal_X* CreateModal_NoShow(class UClass* ModalClass);
class UGFxObject* GetModalObject(class UClass* ModalClass);
bool IsModalActive();
class UGFxModal_X* GetTopModal();
bool CanShowModal();
class UGFxModal_X* ShowModalObject(class UGFxModal_X* Modal);
class UGFxModal_X* CreateModal(class UClass* ModalClass);
void OpenFriendsList();
void ViewProfileDelayed(struct FUniqueNetId ProfileId, class FString Error);
void ViewProfile(struct FUniqueNetId ProfileId);
int32_t GetKeyCodeForAction(struct FName Action, unsigned long bGamepad);
class FString GetKeyForAction(struct FName Action, unsigned long bGamepad);
void ExitToOpenContainer(struct FProductHashID ContainerID);
void LeaveMatch();
void ExitToMainMenu();
void PlaySeason();
void PlayNow();
void OpenScreen(class FString ScreenName);
void OnPrintOpenScreens();
void OnChampionshipTrophyScreenFinished();
void ChampionshipTrophyScreenFinished();
void SpawnChampionshipTrophy();
void SeasonSkipPreMatch();
void RestartMatch();
class UProfile_TA* GetProfile();
void HandleGameSessionEnded();
void HandleGameInvitePassword(class FString Password, unsigned long bCanceled);
void HandleGameInvitePasswordRequired();
void ShowPasswordPrompt(struct FScriptDelegate Callback);
void HandleGameInviteConfirmed(class UGFxModal_X* Modal);
void HandleGameInviteConfirmation(struct FName ConfirmationReason);
void HandleGameInviteAccepted();
void ShowSingleButtonMessage(class FString MessageType);
void HandlePCDestroyed(class APlayerController_X* PC);
void HandleReceivedController(class UPlayer* PlayerRef);
void eventOnStart();
static class UOnlineGame_X* GetOnlineGame();
void EventModalClosed();
void EventModalOpened();
};
```

```

// Class TAGame.__GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x2
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x2 : public UObject
{
public:
class UEpicFriendsConverter_TA*           Converter;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x2");
}

return uClassPointer;
};

void
__GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x2__GFxData_Friends_TA__
CreateConvertPlatformFriendsPrompt_0x1_0x1(class UPrivilegeCheck_X* Check);
void __GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x3(class UGFxModal_X* );
void __GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x1(class UGFxModal_X* );
void __GFxData_Friends_TA__CreateConvertPlatformFriendsPrompt_0x2(class
UPrivilegeCheck_X* Check);
};

// Class TAGame.GFxData_Friends_TA
// 0x0140 (0x0098 - 0x01D8)
class UGFxData_Friends_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FPersonaDataId>      Friends;          // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FPersonaDataId>      PlatformFriends; // 0x00A8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FPersonaDataId>      EpicFriends;     // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FPersonaDataId>      bLocked;         // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class URecentPlayers_TA*           RecentPlayers;    // 0x00D8 (0x0008)
[0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
TArray<class UPersona_TA*>        RecentPlayersToBeAdded; // 0x00E0
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FPersonaDataId>      Recent;          // 0x00F0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FPersonaDataId>      Pending;         // 0x0100 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                           NumOnlineFriendsPlatform; // 0x0110 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

```

```

int32_t NumInGameFriendsPlatform; // 0x0114 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t NumOnlineFriendsEpic; // 0x0118 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t NumInGameFriendsEpic; // 0x011C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long bPlatformFriendsDownloadFailed : 1; // 0x0120
(0x0004) [0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bEpicFriendsDownloadFailed : 1; // 0x0120 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bPromptForPin : 1; // 0x0120 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long bRequirePinForFriends : 1; // 0x0120 (0x0004)
[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bUsePresenceGroups : 1; // 0x0120 (0x0004)
[0x0000000000000002] [0x00000010] (CPF_Const | CPF_Config)
unsigned long bCrossplayEnabled : 1; // 0x0120 (0x0004)
[0x0000000000000000] [0x00000020]
unsigned long bPermissionsInitialized : 1; // 0x0120 (0x0004)
[0x0000000000000000] [0x00000040]
class FString ForgotPinURL; // 0x0128 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UGFxData_FriendsHiddenManager_TA* HiddenManager; // 0x0138
(0x0008) [0x0000000000000000]
class UPersonas_TA* PersonasData; // 0x0140 (0x0008)
[0x0000804000002000] (CPF_Transient)
class UOnlineGame_X* OnlineGame; // 0x0148 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UAccountSettingsComponent_TA* AccountSettings; // 0x0150
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
TArray<uint8_t> DefaultPlatformFriendGroups; // 0x0158 (0x0010)
[0x0000000000404002] (CPF_Const | CPF_Config | CPF_NeedCtorLink)
TArray<uint8_t> DefaultEpicFriendGroups; // 0x0168 (0x0010)
[0x0000000000404002] (CPF_Const | CPF_Config | CPF_NeedCtorLink)
TArray<uint8_t> DefaultAllFriendGroups; // 0x0178 (0x0010)
[0x0000000000404002] (CPF_Const | CPF_Config | CPF_NeedCtorLink)
class UOnlineGameParty_X* Party; // 0x0188 (0x0008)
[0x0000800000002000] (CPF_Transient)
TArray<class UPersona_TA*> PartyMemberPersonas; // 0x0190
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UGFxData_PersonaGroup_TA* PlatformPartyPresenceGroup; //
0x01A0 (0x0008) [0x0000000000000000]
class UGFxData_PersonaGroup_TA* EpicPartyPresenceGroup; // 0x01A8
(0x0008) [0x0000000000000000]
class UOtherPlayerProfile_TA* ObservedProfile; // 0x01B0 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class FString PinGrantPlatformName; // 0x01B8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t PinGrantPlatform; // 0x01C8 (0x0004)
[0x0000000000000000]
class UGFxModal_X* ModalProcessing; // 0x01D0 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Friends_TA");
}

return uClassPointer;
};

void __GFxData_Friends_TA__OnShellSet_0x1(class UEpicConfig_X* EpicConfig);
struct FUniqueNetId __GFxData_Friends_TA__OnFriendsListShown_0x1(struct FPersonaDataId F);
bool __GFxData_Friends_TA__HandlePartiesChanged_0x9(class UPersona_TA* PMember);
bool __GFxData_Friends_TA__HandlePartiesChanged_0x8(class UPersona_TA* PMember);
void __GFxData_Friends_TA__HandlePartiesChanged_0x7(class UPersona_TA* RMember);
bool __GFxData_Friends_TA__HandlePartiesChanged_0x5(class UPersona_TA* Member);
bool __GFxData_Friends_TA__HandlePartiesChanged_0x4(class UPersona_TA* Member);
bool __GFxData_Friends_TA__HandlePartiesChanged_0x3(class UPersona_TA* Member);
class UPersona_TA* __GFxData_Friends_TA__HandlePartiesChanged_0x2(struct FPartyMember OtherPMember);
bool __GFxData_Friends_TA__UpdateRecentPlayerList_0x1(struct FPersonaDataId P);
void __GFxData_Friends_TA__CheckNotificationLimits_0x1(class UGFxModal_X* Modal);
void __GFxData_Friends_TA__HandleClearAllClubInvites_0x1(class UGFxModal_X* Modal);
struct FUniqueNetId __GFxData_Friends_TA__DownloadRecentPlayers_0x1(struct FPersonaDataId P);
void __GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x4(class UNotification_TA* Notification);
void __GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x3(class UNotification_TA* Notification);
struct FUniqueNetId __GFxData_Friends_TA__GetSortedInGameFriendSubset_0x3(class UPersona_TA* Persona);
bool __GFxData_Friends_TA__GetSortedInGameFriendSubset_0x2(class UPersona_TA* Persona);
class UPersona_TA* __GFxData_Friends_TA__GetSortedInGameFriendSubset_0x1(struct FPersonaDataId PDID);
void CreateConvertPlatformFriendsPrompt(class UEpicFriendsConverter_TA* Converter);
void HandlePermissionsDownloaded();
void CheckToPromptToConvertPlatformFriends();
void UpdateFriendInfo();
class FString GetDisplayName(struct FUniqueNetId PlayerID, unsigned long bSingleResult, uint8_t& ForcedRelationships);
class FString GetPreferredDisplayName(struct FUniqueNetId PlayerID, uint8_t ForcedRelationships);
class FString GetPreferredSingleDisplayName(struct FUniqueNetId PlayerID, uint8_t ForcedRelationships);
bool IsFriendInParty(struct FUniqueNetId UniqueId);
TArray<struct FUniqueNetId> GetSortedInGameFriendSubset(TArray<struct FPersonaDataId>& FriendArray);
TArray<struct FUniqueNetId> GetSortedInGameEpicFriends();
TArray<struct FUniqueNetId> GetSortedEpicFriends();
bool ShouldPlatformAlwaysShowEpicFriends();

```

```
TArray<struct FUniqueNetId> GetSortedInGamePlatformFriends();
TArray<struct FUniqueNetId> GetSortedPlatformFriends();
TArray<struct FUniqueNetId> GetSortedOnlineFriends();
TArray<struct FUniqueNetId> GetSortedFriends();
void UpdateFriendAndGroupInfo();
void HandleFriendsListChanged(class UOnlinePlayerFriends_X* FriendsRef, class UError* Error);
int32_t GenerateGroupId();
class UGFxData_PersonaGroup_TA* GetGroup(int32_t GroupIdx);
class UGFxData_PersonaGroup_TA* FindGroup(int32_t GroupDataId);
void ReportRecentPlayer(uint8_t OffenseReasonID, struct FUniqueNetId PlayerID);
void ReportRecentPlayerComplete(unsigned long bSuccess);
void HandleReportRecentPlayerCompleted(class URPC_ReportPlayer_TA* RPC);
void ReportRecentPlayerManyReasons(TArray<int32_t> OffenseReasonIDs, struct FUniqueNetId PlayerID);
void UpdatePersonaData(class UPersona_TA* Persona, struct FOnlineStatus StatusData);
void HandleEpicStatusUpdate(struct FOnlineStatus StatusData);
void HandlePsyNetStatusUpdate(struct FOnlineStatus StatusData);
void HandlePlatformStatusUpdate(struct FOnlineStatus StatusData);
void AddFriendNote(struct FUniqueNetId PlayerID, class FString msg);
void RemoveFriendInternal(class UPersona_TA* UnfriendedPersona, struct FUniqueNetId PlayerID);
void RemoveEpicFriendInternal(struct FUniqueNetId PlayerID);
void RemovePlatformFriendInternal(struct FUniqueNetId PlayerID);
void HandleEpicPlayerUnfriended(struct FUniqueNetId PlayerID);
void RemoveEpicFriend(struct FUniqueNetId PlayerID);
void SwapFriendToGroup_Internal(int32_t NewIdx, struct FUniqueNetId PlayerID);
void SwapFriendToGroup_Presence(struct FUniqueNetId PlayerID);
bool IsFriendInGame(struct FUniqueNetId PlayerID);
void SwapFriendToGroup(int32_t NewIdx, struct FUniqueNetId PlayerID);
void UpdateGroupInfo(struct FUniqueNetId PlayerID);
int32_t GetFriendGroupCount(uint8_t PersonalInfoOrigin);
int32_t GetEpicFriendGroupCount();
int32_t GetPlatformFriendGroupCount();
void ChangeGroupName(int32_t GroupIdx, class FString NewName);
void DeleteFriendGroup(int32_t GroupIdx);
void CreateNewFriendGroup(class FString GroupName);
void HandleDeclineEpicFriendInviteComplete(struct FUniqueNetId PlayerID, class UError* Error);
void DeclineEpicFriendInvite(struct FUniqueNetId PlayerID);
void OnAcceptFriendInviteComplete(struct FUniqueNetId AcceptedId, unsigned long bSuccess,
unsigned long bValidPin, class FString Error);
void HandleAcceptEpicFriendInviteComplete(struct FUniqueNetId AcceptedId, class UError* Error);
void AcceptEpicFriendInvite(struct FUniqueNetId PlayerID, class FString InPin);
void AttemptAcceptEpicFriendInvite(int32_t NotificationID, struct FUniqueNetId PlayerID);
void ShowPinGrantForFriendInvite(int32_t NotificationID, struct FUniqueNetId PlayerID);
void OnReceivedEpicFriendInvite(struct FUniqueNetId PlayerID);
void OnReceivedFriendRequest(struct FUniqueNetId PlayerID);
bool HasExistingFriendRequestNotification(class UFriendRequestNotification_TA* Notification,
struct FUniqueNetId ForPlayerID);
void CreateEpicFriendInviteNotification(class UPersona_TA* InPersona, unsigned long
bSuppressPopup);
void OnCrossPlatformSupportURLClick(class UGFxModal_X* Modal);
void OnFriendInviteSentComplete(struct FUniqueNetId PlayerID, unsigned long bSuccess,
unsigned long bValidPin, class FString Error);
```

```
void OnInviteToEpicFriendsSentSuccess(struct FUniqueNetId PlayerID);
void ShowEpicFriendInviteError(class FString DisplayName, class UError* AddFriendError);
void HandleEpicFriendInviteFailed(struct FUniqueNetId PlayerID, class UError* AddFriendError);
void HandleEpicFriendInviteSucceeded(struct FUniqueNetId PlayerID);
void InviteEpicFriendWithPin(struct FUniqueNetId PlayerID, class FString InPin);
void OnPinInviteFriendNamesReady(struct FUniqueNetId PlayerID, class FString PlayerName,
class FString PlatformName, int32_t Platform);
void InviteEpicFriend(struct FUniqueNetId PlayerID);
bool IsSocialBanned();
void HandleRequestLinkedAccount(unsigned long bSuccess, TArray<struct FLinkedAccountData> ReceivedAccountData);
void HandleQueriedUserByEpicDisplayName(unsigned long bWasSuccessful, class FString
QueriedDisplayName, struct FUniqueNetId QueriedPlayerId);
void InviteToEpicFriendsByName(class FString DisplayName);
void HandleEpicFriendInviteRemoved(struct FUniqueNetId PlayerToRemove);
void HandleIncomingEpicFriendInvite(struct FUniqueNetId RequestingPlayer, class FString
RequestingNick, unsigned long bSuppressPopup);
void HandleExistingIncomingEpicFriendInvite(struct FUniqueNetId RequestingPlayer, class
FString RequestingNick);
void HandleNewIncomingEpicFriendInvite(struct FUniqueNetId RequestingPlayer, class FString
RequestingNick);
void BlockLinkedPersonas();
void HandleBlockedListUpdated(class UOnlinePlayerFriends_X* FriendsObject);
void DownloadBlockedList();
void HandleReceivedRecentPlayerDetails(TArray<struct FOnlineFriend>& OnlineRecentPlayers);
void DownloadRecentPlayersComplete();
void RetryRecentPlayerDetails();
void DownloadRecentPlayers();
void InviteToMatch(struct FUniqueNetId PlayerID);
void HandlePlayerUnblocked(struct FUniqueNetId PlayerID, class UError* RequestError);
void UnblockPlayer(struct FUniqueNetId PlayerID);
void HandlePlayerBlocked(struct FUniqueNetId PlayerID, class UError* RequestError);
void BlockPlayer(struct FUniqueNetId PlayerID);
void HandleClearAllClubInvites();
void CheckNotificationLimits();
void AddBlockedPlayerData(class UPersona_TA* Persona);
void AddPendingPlayerData(class UPersona_TA* Persona);
void AddFriendData(class UPersona_TA* Persona);
void UpdateRecentPlayerList();
void UpdatePlatformPersonaFromEpicId(struct FUniqueNetId PlatformId, struct FUniqueNetId
EpicId);
void HandleRecentPlayerLinkedAccountReceived(unsigned long bSuccess, TArray<struct
FLinkedAccountData> ReceivedAccountData);
void HandlePlayerAddedToMatch(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
void HandleMatchGUIDChanged(class AGRI_X* GRI);
void HandleGameEventRemoved(class AGameEvent_TA* GameEvent);
void HandleGameEventAdded(class AGameEvent_TA* GameEvent);
void HandleCrossPlayChanged();
void HandleAccountSettingsLoaded(class UAccountSettingsComponent_TA*
InAccountSettings);
void HandlePersonaSaveLoaded(class UPersonaSave_TA* PersonaSave);
void HandleProfileSet(class ULocalPlayer_TA* InLocalPlayer);
void HandlePsyNetLogin(class UPsyNetConnection_X* C);
void HandleEpicFriendsListDownloadCompleted(unsigned long bSuccess);
```

```

void HandlePlatformFriendsListDownloadCompleted(unsigned long bSuccess);
void InitGroups();
void SetupPersonaGroup(uint8_t Group, uint8_t GroupOrigin);
class UPersona_TA* GetAndSetPersona(struct FOnlineFriend PersonaData, uint8_t Origin);
class UGFxData_PersonaGroup_TA* CreatePartyPersonaGroup(uint8_t GroupOrigin);
void ValidatePartyPersonaGroup(uint8_t GroupOrigin);
void AddEpicFriendToPartyGroup(class UPersona_TA* PartyMemberPersona);
void AddPlatformFriendToPartyGroup(class UPersona_TA* PartyMemberPersona);
void ClearObservedProfile();
void LoadObservedProfile(struct FUniqueNetId PlayerID);
int32_t GetHiddenPresenceld();
int32_t GetInGamePresenceld();
int32_t GetOfflinePresenceld();
void HandlePartiesChanged(class UOnlineGameParty_X* PartyObject);
class UPersona_TA* GetOrCreatePersona(struct FOnlineFriend PersonaData, uint8_t Origin);
void eventOnRemoved();
void OnEpicFriendThrottlingHit(int32_t WaitSeconds);
void OnPinRequiredForFriendsUpdated(unsigned long bIsRequired);
void HandlePinRequiredForFriendsDownloaded();
void OnFriendsListShown();
void eventOnShellSet();
};

// Class TAGame.__GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x1
// 0x000C (0x0060 - 0x006C)
class U_GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x1 : public UObject
{
public:
    class UPersona_TA*           InPersona;          // 0x0060 (0x0008)
    [0x0000000000000000]
    unsigned long                 bSuppressPopup : 1; // 0x0068 (0x0004)
    [0x0000000000000000] [0x00000001]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x1");
        }
    }

    return uClassPointer;
};

void
__GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x1__GFxData_Friends_TA__Creat
eEpicFriendInviteNotification_0x2_0x2(class UNotification_TA* Notification);
void
__GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x1__GFxData_Friends_TA__Creat
eEpicFriendInviteNotification_0x2_0x1(class UNotification_TA* Notification);
void __GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x2(class UPersona_TA* P);

```

```
bool __GFxData_Friends_TA__CreateEpicFriendInviteNotification_0x1(class UNotification_TA* N);
};

// Class TAGame.FriendRequestNotification_TA
// 0x0048 (0x0170 - 0x01B8)
class UFriendRequestNotification_TA : public UNotification_TA
{
public:
    struct FUniqueNetId           PlayerID;           // 0x0170 (0x0048)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FriendRequestNotification_TA");
        }

        return uClassPointer;
    };

    bool ShouldShow();
    struct FUniqueNetId GetSenderID();
    class UFriendRequestNotification_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.__GFxData_Friends_TA__DeleteFriendGroup_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GFxData_Friends_TA__DeleteFriendGroup_0x1 : public UObject
{
public:
    int32_t                  TargetPresenceId;          // 0x0060 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Friends_TA__DeleteFriendGroup_0x1");
        }

        return uClassPointer;
    };

    void __GFxData_Friends_TA__DeleteFriendGroup_0x1(class UPersona_TA* P);
};
```

```

// Class TAGame.__GFxData_Friends_TA__HandleMatchGUIDChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Friends_TA__HandleMatchGUIDChanged_0x1 : public UObject
{
public:
class UPersona_TA* Persona; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Friends_TA__HandleMatchGUIDChanged_0x1");
}

return uClassPointer;
};

bool __GFxData_Friends_TA__HandleMatchGUIDChanged_0x1(class APlayerReplicationInfo* P);
};

// Class TAGame.__GFxData_Friends_TA__HandlePartiesChanged_0x1
// 0x0050 (0x0060 - 0x00B0)
class U__GFxData_Friends_TA__HandlePartiesChanged_0x1 : public UObject
{
public:
struct FUniqueNetId PrimaryPlayerId; // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
class UOnlineGameParty_TA* PartyObjectTA; // 0x00A8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Friends_TA__HandlePartiesChanged_0x1");
}

return uClassPointer;
};

void __GFxData_Friends_TA__HandlePartiesChanged_0x6(struct FUniqueNetId PlayerID);
bool __GFxData_Friends_TA__HandlePartiesChanged_0x1(struct FPartyMember PMember);
};

// Class TAGame.OnlineGameParty_TA

```

```

// 0x00D0 (0x0440 - 0x0510)
class UOnlineGameParty_TA : public UOnlineGameParty_X
{
public:
TArray<class UPartyMemberLoadout_TA*>      PartyLoadouts;           // 0x0440
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                           MaxPartyMessages;          // 0x0450 (0x0004)
[0x0000004000000000]
class FString                         PlayerJoinPartyString;    // 0x0458 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                         PlayerLeavePartyString;   // 0x0468 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                         LocalPlayerLeavePartyString; // 0x0478 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                         PlayerNewLeaderPartyString; // 0x0488 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UProductAssetCache_TA*          AssetCache;             // 0x0498 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
unsigned long                        bIsVerifyingProducts : 1; // 0x04A0 (0x0004)
[0x0001000000000000] [0x00000001]
class UEOSVoiceManager_TA*           VoiceManager;          // 0x04A8
(0x0008) [0x0001800000000000]
struct FScriptDelegate              __EventPartyMemberLoadoutChange__Delegate; // 0x04B0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              __EventPartyMemberProfileChanged__Delegate; // 0x04C8 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              __EventPartyMemberLoadoutRemoved__Delegate; // 0x04E0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              __EventPartyMemberLeftMatch__Delegate;    // 0x04F8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameParty_TA");
}

return uClassPointer;
};

void __OnlineGameParty_TA__OnInit_0x3(struct FUniqueNetId _);
void __OnlineGameParty_TA__OnInit_0x2(class UOnlineGameParty_X* _);
void __OnlineGameParty_TA__OnInit_0x1(class UOnlineMessageComponent_X* _, class UPartyMessage_Chat_TA* M);
void __OnlineGameParty_TA__HandleProfileSet_0x4(class APlayerControllerBase_TA* NewController);
void __OnlineGameParty_TA__HandleProfileSet_0x5(class UAccountSettingsComponent_TA* Settings);
void __OnlineGameParty_TA__HandleProfileSet_0x1(class UProfileLoadoutSave_TA* LoadoutSave);

```

```
void __OnlineGameParty_TA__BroadcastXPLevelUpdateMessage_0x2(class UClientXPSave_TA* XPSave);
void __OnlineGameParty_TA__BroadcastXPLevelUpdateMessage_0x1(class UClientXPSave_TA* XPSave);
struct FUniqueNetId __OnlineGameParty_TA__GetPartyMessageError_0x2(struct FPartyMember R);
bool __OnlineGameParty_TA__GetPartyMessageError_0x1(struct FPartyMember P);
class UPartyMemberLoadout_TA* GetLoadout(struct FUniqueNetId PlayerID);
class UError* GetKickedFromPartyError(uint8_t Reason);
void CheckForJoiningPlayerKick(struct FUniqueNetId PlayerID);
class UError* CheckForPlayerInviteError(struct FUniqueNetId PlayerID);
class UError* GetJoinPartyError(struct FActiveLobbyInfo& LobbyInfo);
void HandleCrossPlatformChatStateChanged();
bool CanBroadcastMatchmakingMessages();
bool AllowSplitScreenPlayer(int32_t ControllerId, class UError*& OutError);
bool CanPlayersCrossTextChat(int32_t SenderIndex);
int32_t FindLocalPartyPlayer();
void HandleLobbyMessageFailed(class UError* Error);
void HandleChatMessage(struct FUniqueNetId PlayerID, class FString Text);
class UError* GetPartyMessageError(struct FUniqueNetId PlayerID);
void SendPartyLobbyMessage(class FString Message);
void UpdateOnlineProducts(class UPartyMemberLoadout_TA*& Loadout);
void HandleCrossplayEnabledChanged(class UAccountSettingsComponent_TA* AccountSettings);
void UpdateAssetCache();
void HandleGameEventFinished(class AGameEvent_TA* GameEvent);
bool ClearServersForPostGameRankedMatch();
void HandlePlayerRemoved(class AGameEvent_TA* InGameEvent, class APRI_TA* InPRI);
bool WantsToFollowTheLeaderOutOfGame();
int32_t BuildMatchmakingRestrictions();
void HandlePrimaryPlayerSet(class APlayerController_Menu_TA* PC);
void RequestRejoinPartyInfo();
void BroadcastPlayerLoadout(class UPartyMemberLoadout_TA* MemberLoadout);
static TArray<struct FOnlineProductData> GetProfileProducts(class UProfile_TA* Profile);
static class UPartyMemberLoadout_TA* CreatePartyMemberLoadout(class UProfile_TA* Profile, class ULocalPlayer_TA* Player);
void BroadcastLocalXPLevelUpdateDelayed();
void BroadcastLocalPlayerXPLevelUpdate();
void BroadcastLocalPlayerLoadoutsDelayed();
void BroadcastLocalPlayerLoadouts();
void HandleVoiceRoomChanged(class UOnlinePlayer_X* OnlinePlayer, class FString RoomName, unsigned long bSuccess);
void HandleVoiceManager();
void BroadcastPlayerVoiceRoom(class UOnlinePlayer_X* OnlinePlayer);
void BroadcastLocalPlayerVoiceRooms();
void BroadcastAllLocalPlayers();
void HandlePlayerRejoined(class UOnlineMessageComponent_X* Component, class UObject* Message);
static void UpdatePartyMemberLoadout(int32_t LoadoutIndex, TArray<class UPartyMemberLoadout_TA*>& MemberLoadouts, TArray<struct FOnlineProductData>& NewProducts);
void HandleProfileChangeMessage(class UOnlineMessageComponent_X* Component, class UObject* Message);
bool IsPrimaryMemberForPlayer(struct FUniqueNetId PrimaryPlayerId, struct FUniqueNetId
```

```

UserId);
void HandleLoadoutMessage(class UOnlineMessageComponent_X* Component, class UObject* Message);
void UpdateXPLevelForPlayer(struct FUniqueNetId InPlayerId, int32_t InXPLevel);
void HandleXPLevelUpdateMessage(class UOnlineMessageComponent_X* InComponent, class UPartyMessage_XPLevelUpdate_TA* InMessage);
void SendXPUpdateMessage(class ULocalPlayer_TA* InLocalPlayer, int32_t InXPLevel);
void HandleXPLevelChanged(class UClientXPSave_TA* InXPSave, int32_t InNewLevel, int32_t InOldLevel);
void BroadcastXPLevelUpdateMessage(class ULocalPlayer_TA* InLocalPlayer);
void HandlePreviewTeamChange(class UProfileLoadoutSave_TA* LoadoutSave);
void BroadcastPendingLocalPlayerLoadouts();
void BroadcastPartyInfo();
void BroadcastStateDelayed();
void HandleLoadoutChange(class ULoadout_TA* Loadout);
void BroadcastProfileProductChanges();
void HandleProfileChange(class ULocalPlayer_TA* Player);
void OnPartyLeaderChanged();
void OnNewLobby();
void RemoveAllPartyMembers();
void RemovePartyMemberByIDWithNotify(struct FUniqueNetId MemberId);
void RemovePartyMemberByID(struct FUniqueNetId MemberId);
bool LeaveParty(class FString Reason);
void HandleLocalPlayerLeave(class ULocalPlayer* Player);
void OnEditExistingPartyMember(int32_t MemberIdx);
void OnAddNewPartyMember(int32_t MemberIdx);
void RefreshLoadouts(int32_t MemberIdx);
void RefreshLoadoutForPlayer(class ULocalPlayer_TA* LocalPlayer);
void ClearPartyChat();
void AddPartyServiceLobbyMessage(class FString Message);
void AddPlayerPartyServiceLobbyMessage(struct FPartyMember Member, class FString Message);
int32_t LocalPlayerSort(class UPartyMemberLoadout_TA* Left, class UPartyMemberLoadout_TA* Right);
void SetLoadout(class ULocalPlayer* Player, class UPartyMemberLoadout_TA*& Loadout);
void HandleProfileSet(class ULocalPlayer_TA* P);
void OnInit();
void EventPartyMemberLeftMatch(struct FUniqueNetId PartyMemberID);
void EventPartyMemberLoadoutRemoved();
void EventPartyMemberProfileChanged(struct FUniqueNetId PlayerID);
void EventPartyMemberLoadoutChange(struct FUniqueNetId PlayerID);
};

// Class TAGame.RecentPlayers_TA
// 0x0070 (0x0070 - 0x00E0)
class URecentPlayers_TA : public UComponent
{
public:
TArray<struct FPersonaDataId> RecentList; // 0x0070 (0x0010)
[0x0008004000400000] (CPF_NeedCtorLink)
TArray<struct FPlayerGameID> RecentGameIDs; // 0x0080
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPlayerLoadoutData> ObservedLoadouts; // 0x0090
(0x0010) [0x0008004000400000] (CPF_NeedCtorLink)

```

```
class UPersonaSave_TA*           PersonaSave;          // 0x00A0 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t                         MaxRecentPlayerCount; // 0x00A8 (0x0004)
[0x0000004000002000] (CPF_Transient)
struct FScriptDelegate           __RecentList_ChangeNotify; // 0x00B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __ObservedLoadouts_ChangeNotify; // 0x00C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RecentPlayers_TA");
}

return uClassPointer;
};

bool __RecentPlayers_TA__UpdateFromOnline_0x2(struct FPlayerLoadoutData L);
struct FPersonaDataId __RecentPlayers_TA__UpdateFromOnline_0x1(struct FOnlineFriend RP);
void __RecentPlayers_TA__SetPersonaSave_0x1();
bool __RecentPlayers_TA__OnRecentsChanged_0x1(struct FPlayerGameID G);
void __ObservedLoadouts__ChangeNotifyFunc();
void __RecentList__ChangeNotifyFunc();
void AddPlayerInternal(class UPersona_TA* Persona, class UVanityLoadout_TA*
VanityLoadout);
bool ShouldAddToRecentPlayerList(class UPersona_TA* Persona, unsigned long
bIncludeFriends);
void OnRecentsChanged();
void SetPersonaSave(class UPersonaSave_TA* InPersonaSave);
bool IsRecentPlayer(struct FPersonaDataId InRecentPlayer, class UPersonas_TA*
InPersonasData);
void FilterFriends(class UPersonas_TA* PersonasData);
void UpdateFromOnline(TArray<struct FOnlineFriend>& OnlineRecents);
class FString GetGameIDPlayedWith(struct FUniqueNetId PlayerID);
void UpdatePlayerVanity(struct FUniqueNetId PlayerID, class UVanityLoadout_TA* NewerVanity);
void AddMatchMate(class UPersona_TA* Persona, class UVanityLoadout_TA* Loadout, class
FString InGameID);
void AddPlayer(class UPersona_TA* Persona, class UVanityLoadout_TA* Loadout);
};

// Class TAGame.IVanityLoadout_TA
// 0x0000 (0x0060 - 0x0060)
class UVanityLoadout_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IVanityLoadout_TA");
}

return uClassPointer;
};

struct FOnlineProductData GetAvatarBorderData();
struct FOnlineProductData GetBannerData();
};

// Class TAGame.__GFxData_Friends_TA__HandlePlayerAddedToMatch_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Friends_TA__HandlePlayerAddedToMatch_0x1 : public UObject
{
public:
class APRI_TA*           PRI;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Friends_TA__HandlePlayerAddedToMatch_0x1");
}

return uClassPointer;
};

void __GFxData_Friends_TA__HandlePlayerAddedToMatch_0x1(uint8_t InVanityType);
};

// Class TAGame.__GFxData_Friends_TA__HandlePlayerBlocked_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__GFxData_Friends_TA__HandlePlayerBlocked_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;        // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Friends_TA_HandlePlayerBlocked_0x1");
}

return uClassPointer;
};

void __GFxData_Friends_TA_HandlePlayerBlocked_0x1(class UGFxModal_X* Modal);
};

// Class TAGame._GFxData_Friends_TA_IsFriendInParty_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxData_Friends_TA_IsFriendInParty_0x1 : public UObject
{
public:
struct FUniqueNetId           Uniqueld;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Friends_TA_IsFriendInParty_0x1");
}

return uClassPointer;
};

bool __GFxData_Friends_TA_IsFriendInParty_0x1(struct FPartyMember PM);
};

// Class TAGame._GFxData_Friends_TA_UnblockPlayer_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxData_Friends_TA_UnblockPlayer_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Friends_TA_UnblockPlayer_0x1");
}
}
```

```

return uClassPointer;
};

void __GFxData_Friends_TA_UnblockPlayer_0x1(class UGFxModal_X* Modal);
};

// Class TAGame.__GFxData_Garage_TA_SetPreviewProduct_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Garage_TA_SetPreviewProduct_0x1 : public UObject
{
public:
int32_t SlotIndex; // 0x0060 (0x0004)
[0x0000000000000000]
struct FProductHashID HashID; // 0x0064 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Garage_TA_SetPreviewProduct_0x1");
}

return uClassPointer;
};

void __GFxData_Garage_TA_SetPreviewProduct_0x1();
};

// Class TAGame.GFxData_Garage_TA
// 0x0210 (0x0098 - 0x02A8)
class UGFxData_Garage_TA : public UGFxDataSingleton_X
{
public:
TArray<class UProductSlot_TA*> LoadoutSlots; // 0x0098 (0x0010)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class ACarPreviewActor_TA* PreviewActor; // 0x00A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
TArray<class UGFxData_GarageProductSlot_TA*> GarageProductSlots; // 
0x00B0 (0x0010) [0x000000000400000] (CPF_NeedCtorLink)
int32_t TabIndex; // 0x00C0 (0x0004)
[0x000000004000000] (CPF>EditInlineNotify)
TArray<struct FGarageTeam> Teams; // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t PaintFinishSlotIndex; // 0x00D8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t PreviewTeam; // 0x00DC (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
TArray<struct FGFxTeamColor> CustomColors; // 0x00E0
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)

```

```

TArray<struct FGFxTeamColor> Team0Colors; // 0x00F0
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor> Team1Colors; // 0x0100
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bTeam0FinishDisabled : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bTeam1FinishDisabled : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bCustom0FinishDisabled : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long bCustom1FinishDisabled : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bTeam0ColorForced : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long bTeam1ColorForced : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000020] (CPF_EditInlineNotify)
unsigned long bCustom0ColorForced : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000040] (CPF_EditInlineNotify)
unsigned long bCustom1ColorForced : 1; // 0x0110 (0x0004)
[0x0000000040000000] [0x00000080] (CPF_EditInlineNotify)
class FString ForcedTeam0FinishName; // 0x0118 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ForcedTeam1FinishName; // 0x0128 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t ForcedTeam0Color; // 0x0138 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ForcedTeam1Color; // 0x013C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ForcedCustom0Color; // 0x0140 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ForcedCustom1Color; // 0x0144 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UProfile_TA* Profile; // 0x0148 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FLoadoutData Loadouts[0x2]; // 0x0150 (0x0080)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FLoadoutData LastUnforcedLoadout[0x2]; // 0x01D0
(0x0080) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class ULoadoutSet_TA* LoadoutSet; // 0x0250 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UCarPreviewSet_TA* CarPreviewSet; // 0x0258 (0x0008)
[0x0000000000000000]
class UExplosionPreviewer_TA* ExplosionPreview; // 0x0260
(0x0008) [0x00000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FName GoalExplosionSeqName; // 0x0268 (0x0008)
[0x0000000000000002] (CPF_Const)
class UProductSlot_TA* CurrentSlot; // 0x0270 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FScriptDelegate __PreviewTeamAction__Delegate; // 0x0278
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventResetTabIndexes__Delegate; // 0x0290
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

public:

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Garage_TA");
}

return uClassPointer;
};

void __GFxData_Garage_TA__InitProductSlots_0x1(class UGFxData_GarageProductSlot_TA* S);
bool __GFxData_Garage_TA__GetPrimaryPlayer_0x1(class ULocalPlayer_TA* LP);
class UOnlineProductStoreSet_TA* GetLocalOnlineProductSet();
void UpdateExplosionPreviewer();
void HandleUpdatedNumLocalPlayers(class UGameViewportClient_TA* GVC, int32_t LocalPlayerCount);
bool AllowTeamSpecificEquip(int32_t SlotIndex, struct FProductHashID HashID);
bool AreDuplicateProducts(struct FProductInstanceId ProductA, struct FProductInstanceId ProductB);
void SetPreviewLoadout(struct FLoadoutData& InLoadout);
bool HasNewProducts();
void UpdateColorSettings();
static bool InitForcedCustomColor(int32_t TeamIndex, class UProductAsset_Body_TA* BodyAsset, class UProductAsset_Skin_TA* SkinAssets, int32_t& OutColor);
static bool InitForcedTeamColor(int32_t TeamIndex, class UProductAsset_Body_TA* BodyAsset, class UProductAsset_Skin_TA* SkinAssets, int32_t& OutColor);
static class FString GetForcedFinishLabel(unsigned long bFinishDisabled, class UProductAsset_Body_TA* BodyAsset, class UProductAsset_Skin_TA* SkinAsset, int32_t TeamID);
void OnProductsChanged();
void HandleDLCChange();
void HandleLoadoutChanged(class ULoadoutSet_TA* InLoadoutSet, class ULoadout_TA* InLoadout);
void CommitLoadout();
void SetPreviewTeam(int32_t Team);
void UpdateCustomFinish(int32_t Team, struct FProductHashID HashID);
void SetCustomFinish(int32_t Team, struct FProductHashID HashID);
bool IsCustomFinishDisabled(int32_t Team);
void SetPreviewCustomFinish(struct FProductHashID HashID);
void UpdateTeamFinish(int32_t Team, struct FProductHashID HashID);
void SetTeamFinish(int32_t Team, struct FProductHashID HashID);
bool IsTeamFinishDisabled(int32_t Team);
void SetPreviewTeamFinish(int32_t Team, struct FProductHashID HashID);
void SetCustomColor(int32_t Team, int32_t ColorID);
bool IsCustomColorDisabled(int32_t Team);
void SetPreviewCustomColor(int32_t ColorID);
void SetTeamColor(int32_t Team, int32_t ColorID);
bool IsTeamColorDisabled(int32_t Team);
void SetPreviewTeamColor(int32_t Team, int32_t ColorID);
void ClearPreviewPaint(int32_t Team);
void EquipProduct(int32_t SlotIndex, struct FProductHashID HashID, uint8_t EquipAction);
void UpdateLastUnforcedLoadout(int32_t SlotIndex, struct FProductHashSource ProductSource,

```

```

uint8_t EquipAction);
TArray<int32_t> GetEquipActionAsTeamIndex(uint8_t EquipAction);
void OnEquipBodyProduct(int32_t InProductID);
bool BodyHasForcedProducts(class UProduct_TA* ProductBody);
bool BodyIsLicensedOrForced(class UProduct_TA* InBody);
void ClearPreviewLoadout(int32_t PendingPreviewProductID);
void ClearPreviewProduct();
void SetPreviewProduct(int32_t SlotIndex, struct FProductHashID HashID);
void SetTabIndex(int32_t InTabIndex);
void ResetAllLocalPlayerTabIndexes();
void ResetTabIndex();
class ULocalPlayer_TA* GetPrimaryPlayer();
void SetSlot(int32_t SlotIndex);
void InitColors();
void InitProductSlots();
void InitTeams();
void InitLastUnforcedLoadout();
void InitLoadouts(class ULoadoutSet_TA* NewLoadoutSet, int32_t TeamIndex);
void HandleLoadoutSaveLoaded(class UProfileLoadoutSave_TA* LoadoutSave);
void LoadUserLoadout();
void HandlePreviewActorsChanged(class UCarPreviewSet_TA* G);
void eventOnRemoved();
void eventOnShellSet();
void EventResetTabIndexes();
void PreviewTeamAction();
};

// Class TAGame.__GFxData_Leaderboards_TA__PrefetchAvatars_0x1
// 0x0001 (0x0060 - 0x0061)
class U__GFxData_Leaderboards_TA__PrefetchAvatars_0x1 : public UObject
{
public:
    uint8_t OurPlatform; // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Leaderboards_TA__PrefetchAvatars_0x1");
        }
    }

    return uClassPointer;
};

bool __GFxData_Leaderboards_TA__PrefetchAvatars_0x1(struct FLeaderboardData Entry);
};

// Class TAGame.GFxData_Leaderboards_TA
// 0x0050 (0x0098 - 0x00E8)

```

```

class UGFxData_Leaderboards_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FLeaderboardData> LeaderboardDataSet; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t PlayerLeaderboardValue; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t PlayerLeaderboardDivision; // 0x00AC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float PlayerLeaderboardMMR; // 0x00B0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString PlayerLeaderboardName; // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bShowGlobalLeaderboards : 1; // 0x00C8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bWordFilterError : 1; // 0x00C8 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
class UOnlineGameLeaderboards_X* Leaderboards; // 0x00D0
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FName PendingLeaderboardRequestID; // 0x00D8
(0x0008) [0x0000000000002000] (CPF_Transient)
class UVanitySetManager_TA* VanityManager; // 0x00E0
(0x0008) [0x0000800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Leaderboards_TA");
}

return uClassPointer;
};

struct FUniqueNetId __GFxData_Leaderboards_TA__PrefetchAvatars_0x2(struct
FLeaderboardData FilteredEntry);
void ViewProfile(int32_t InRowIndex);
void HandleSanitizeError(class FString OriginalPhrase, class UError* Error);
void HandlePlayerNameSanitized(class FString Original, class FString Sanitized);
void SanitizePlayerNames(TArray<struct FLeaderboardData>& Table);
void GetLeaderboardFriends(struct FName LeaderboardId);
void HandlePlayerFriendLeaderboardRecieved(class UOnlineGameLeaderboards_X*
LeaderboardsRef, struct FName LeaderboardId, class UError* Error, struct FLeaderboardData&
PlayerLeaderboardData);
void HandlePlayerLeaderboardRecieved(class UOnlineGameLeaderboards_X* LeaderboardsRef,
struct FName LeaderboardId, class UError* Error, struct FLeaderboardData&
PlayerLeaderboardData);
void RequestClearPendingAvatarDownloads();
void PrefetchAvatars(TArray<struct FLeaderboardData>& InData);
void HandleLeaderboardReceived(struct FName NewLeaderboardID, TArray<struct
FLeaderboardData>& NewLeaderboardData);

```

```

void OnLeaderboardError(class FString LeaderboardError);
void OnLeaderboardLoaded();
void HandleLeaderboardError(class UError* Error);
TArray<struct FLeaderboardRequestInfo> GetDisplayedRankedLeaderboards();
void GetLeaderboard(struct FName LeaderboardName);
void SetShowGlobalLeaderboards(unsigned long bGlobal);
void InitShowGlobalLeaderboards();
void GetPlayerLeaderboardValue(struct FName LeaderboardName, struct FScriptDelegate NewDelegate);
void BeginGetLeaderboard(struct FName LeaderboardName);
void eventOnShellSet();
class UOnlineGame_X* GetOnlineGame();
};

// Class TAGame.__GFxData_LocalPlayer_TA__HandleXPLoaded_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_LocalPlayer_TA__HandleXPLoaded_0x1 : public UObject
{
public:
class UClientXPSave_TA*           XPSave;                      // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_LocalPlayer_TA__HandleXPLoaded_0x1");
}

return uClassPointer;
};

void __GFxData_LocalPlayer_TA__HandleXPLoaded_0x1(class USaveData_TA* SaveData);
};

// Class TAGame.ClientXPSave_TA
// 0x0040 (0x00C8 - 0x0108)
class UClientXPSave_TA : public USaveObject_TA
{
public:
int32_t                           TotalXP;                     // 0x00C8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                           Level;                       // 0x00CC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                           CurrentLevelXPTreshold; // 0x00D0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                           NextLevelXPTreshold;   // 0x00D4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FScriptDelegate           __EventXPUpdated__Delegate; // 0x00D8
(0x0018) [0x000000000000400000] (CPF_NeedCtorLink)

```

```

struct FScriptDelegate           __EventLevelUpdated__Delegate;           // 0x00F0
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClientXPSave_TA");
}

return uClassPointer;
};

class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
static void StaticGetXPIInfo(int32_t XPTotal, int32_t& Out_XPLevel, class FString& Out_XPTitle,
int32_t& Out_XPProgressInCurrentLevel, int32_t& Out_XPRequiredForNextLevel);
void GiveXPDrop(struct FOnlineXPReward& XPDrop);
void GetXPIInfo(int32_t& Out_XPTotal, int32_t& Out_XPLevel, class FString& Out_XPTitle, int32_t&
Out_XPProgressInCurrentLevel, int32_t& Out_XPRequiredForNextLevel);
void SetLevelInfo(int32_t InCurrentLevel, int32_t InCurrentLevelXPTreshold, int32_t
InNextLevelXPTreshold);
void SetXPIInfo(struct FXPIInfo& InXPIInfo);
void SetXPIInfoNoNotify(struct FXPIInfo& InXPIInfo);
void SetTotalXP(int32_t InTotalXP);
bool InitLevelInfoNoNotify(int32_t InCurrentLevel, int32_t InCurrentLevelXPTreshold, int32_t
InNextLevelXPTreshold, int32_t& OldLevel);
bool SetTotalXPNoNotify(int32_t InTotalXP, int32_t& OldXP);
void EventLevelUpdated(class UClientXPSave_TA* XPSave, int32_t NewLevel, int32_t OldLevel);
void EventXPUpdated(class UClientXPSave_TA* XPSave, int32_t OldXP);
};

// Class TAGame.SaveData_TA
// 0x01D8 (0x0060 - 0x0238)
class USaveData_TA : public UObject
{
public:
class USaveGameManager_TA*           SaveManager;           // 0x0060
(0x0008) [0x0000000000002000] (CPF_Transient)
unsigned long                      bFromPreviousSave : 1;           // 0x0068 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long                      bEnableOnlineData : 1;           // 0x0068 (0x0004)
[0x0000000000006000] [0x00000002] (CPF_Transient | CPF_Config)
unsigned long                      bUnloaded : 1;           // 0x0068 (0x0004)
[0x0000008000002000] [0x00000004] (CPF_Transient)
TArray<class UOnlineProduct_TA*>   OnlineProducts;           // 0x0070
(0x0010) [0x0008008000400000] (CPF_NeedCtorLink)
class UOnlineProductStoreSet_TA*    OnlineProductSet;           // 0x0080
(0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
CPF_EditInline)
class UProductTransactions_TA*     ProductTransactions;           // 0x0088
(0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
```

```

CPF>EditInline)
TArray<class UProfile_TA*> Profiles; // 0x0090 (0x0010)
[0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)
TArray<float> XPGainPercentage; // 0x00A0 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
int32_t CurrentLevel; // 0x00B0 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t SchematicResources; // 0x00B4 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t SaveDataVersion; // 0x00B8 (0x0004)
[0x0000000000000000]
class UObjectProvider* SaveObjectProvider; // 0x00C0 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
TArray<class USaveObject_TA*> SaveObjects; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t SaveTick; // 0x00D8 (0x0004)
[0x0000000000000000]
class UAsyncTask* SyncOnlineStorageTask; // 0x00E0 (0x0008)
[0x0001000000002000] (CPF_Transient)
class UOnlineStorageSyncManager_TA* StorageSyncManager; // 0x00E8
(0x0008) [0x0000008000002000] (CPF_Transient)
class UEOSOnlineStorageUploader_TA* EOSStorageUploader; // 0x00F0
(0x0008) [0x0000008000002000] (CPF_Transient)
class UOnlineProductExpirationHelper_TA* ExpirationHelper; // 0x00F8
(0x0008) [0x0000008000002000] (CPF_Transient)
class UCachedUnlockedProducts_TA* CachedUnlockedProducts; //
0x0100 (0x0008) [0x0000008000002000] (CPF_Transient)
class UOnlineDLCProductCache_TA* OnlineDLCProductCache; // 0x0108
(0x0008) [0x0000004000002000] (CPF_Transient)
TArray<class UOnlineProduct_TA*> NewProductNotificationQueue; //
0x0110 (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString RP_rewardForCurrentSeason; // 0x0120 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString RP_rewardForOldSeasons; // 0x0130 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString CreditRefundForCurrentSeason; // 0x0140 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString CreditRefundForOldSeasons; // 0x0150 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FScriptDelegate __EventOnlineXPIinitialized__Delegate; // 0x0160
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventNewOnlineProduct__Delegate; // 0x0178
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventCrewRocketPassAcquired__Delegate; //
0x0190 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventRemovedOnlineProduct__Delegate; // 0x01A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventOnlineProductsSynced__Delegate; // 0x01C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventOnlineSavedDataSynced__Delegate; // 0x01D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSchematicResourcesChanged__Delegate; //
0x01F0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventRedeemedOnlineProducts__Delegate; //

```

0x0208 (0x0018) [0x0000000000400000] (CPF\_NeedCtorLink)  
struct FScriptDelegate                    \_\_OnlineProducts\_\_ChangeNotify;                // 0x0220  
(0x0018) [0x0000000000400000] (CPF\_NeedCtorLink)

public:  
static UClass\* StaticClass()  
{  
static UClass\* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SaveData\_TA");  
}  
  
return uClassPointer;  
};

void \_\_SaveData\_TA\_\_HandleDataLoaded\_0x2(class UClass\* T);  
void \_\_SaveData\_TA\_\_HandleDataLoaded\_0x1(class UOnlineProduct\_TA\* Op);  
void \_\_SaveData\_TA\_\_HandleProfileSet\_0x1(class USaveObject\_TA\* S);  
void \_\_SaveData\_TA\_\_ValidateLoadouts\_0x1(class UProfile\_TA\* P);  
bool \_\_SaveData\_TA\_\_GetNewProducts\_0x1(class UProduct\_TA\* P);  
bool \_\_SaveData\_TA\_\_GiveXPDrops\_0x1(struct FOnlineXPReward Drop);  
bool \_\_SaveData\_TA\_\_GetOnlineProductQuantity\_0x4(struct FProfileProduct PP);  
bool \_\_SaveData\_TA\_\_GetOnlineProductQuantity\_0x1(struct FProfileProduct P);  
int32\_t \_\_SaveData\_TA\_\_GetUniqueSeriesIDs\_0x1(class UOnlineProduct\_TA\* P);  
class USaveObject\_TA\* \_\_SaveData\_TA\_\_InstantiateSaveObjects\_0x2(class UClass\* T);  
bool \_\_SaveData\_TA\_\_InstantiateSaveObjects\_0x1(class UClass\* T);  
bool \_\_SaveData\_TA\_\_GetEquippedOnlineProducts\_0x1(struct FProductInstanceId Id);  
void \_\_OnlineProducts\_\_ChangeNotifyFunc();  
TArray<struct FProductInstanceId> GetEquippedInstanceIDs();  
TArray<class UOnlineProduct\_TA\*> GetEquippedOnlineProducts();  
void FilterEquippedOnlineProducts(TArray<class UOnlineProduct\_TA\*>& OutOnlineProducts);  
class UOnlinePlayer\_TA\* GetOnlinePlayer();  
void HandleGameplaySettingsSync(class UGameplaySettingsSave\_TA\* GameplaySave);  
void UpdateVideoSettings(class UVideoSettingsSave\_TA\* Settings);  
void HandleVideoSettingsSync(class UVideoSettingsSave\_TA\* VideoSave);  
void UpdateSaveObjectReference(class USaveObject\_TA\* NewObject);  
TArray<class UClass\*> InstantiateSaveObjects();  
void SortByTradeHold(unsigned long blsAscending, TArray<class UOnlineProduct\_TA\*>& OutItemList);  
TArray<int32\_t> GetUniqueSeriesIDs();  
class UOnlineProduct\_TA\* FindOnlineProductWithNoAttributesFromID(int32\_t InProductID,  
TArray<class UClass\*>& AllowedAttributes);  
class UOnlineProduct\_TA\* FindFirstOnlineProduct(int32\_t InProductID);  
TArray<class UOnlineProduct\_TA\*> GetUniqueOnlineProducts();  
TArray<class UOnlineProduct\_TA\*> GetAllOnlineProducts();  
int32\_t GetOnlineProductQuantity(class UOnlineProduct\_TA\* Product, unsigned long  
blgnoreEquipped);  
int32\_t GetOnlineProductTradeInQuantity(class UOnlineProduct\_TA\* Product, unsigned long  
blgnoreEquipped, unsigned long bHoldXEInstance);  
TArray<struct FOnlineProductData> InstanceReplicatedOnlineProductData(TArray<struct  
FProductInstanceId>& InstanceIDs);  
bool OwnsProductID(int32\_t InProductID);

```
TArray<class UOnlineProduct_TA*> GetOnlineProducts(int32_t InProductID);
void RemoveNewProducts(struct FProductInstanceID InstanceID);
void TickNewProductNotificationQueue(float _);
int32_t NewProductsCompare(class UOnlineProduct_TA* L, class UOnlineProduct_TA* R);
void HandleOnlineProductsReceived(class URPC_ProductsPlayerGet_TA* RPC);
void FetchOnlineProducts();
void ClearOfflineProducts();
void ClearOnlineProducts();
void SetSchematicResouces(int32_t Total);
void HandleSchematicResourcesReceived(class URPC_SchematicResourcesGet_TA* RPC);
void SyncSchematicResources();
void SyncOnlineProducts();
void GiveXPDrops(TArray<struct FOnlineXPReward>& XPDrops);
void GiveChallengeDrop(struct FOnlineReward ChallengeReward);
void GiveRewardDrop(struct FOnlineXPReward Drop);
void HandleRewardDropNotification(class UPsyNetService_RewardDropReceived_TA* Service);
void HandleOnlineXPIinitialized(class URPC_XPRequest_TA* RPC);
bool CanUseOnlineData(struct FUniqueNetId PlayerID);
void MarkReplayEditorOpened();
void MarkCreditsViewed();
float GetUnlockProgress();
float GetGameModeUnlockProgress();
float GetItemUnlockProgress();
void UpdateGameProgress();
void PrintDebugInfo(class UDebugDrawer* Drawer);
void ValidateLastUnlockDisplayId();
static bool IsProductMoreNew(class UOnlineProduct_TA* InProduct, struct FProductInstanceID InstanceID, uint64_t AddedTimestamp);
void SetOnlineDropDisplayed(struct FProductInstanceID ItemID, uint64_t AddedTime);
void SetItemDropDisplayed(struct FProductInstanceID ItemID, uint64_t ItemAddedTime);
bool OnNewOnlineProductNoNotify(class UOnlineProduct_TA* OnlineProduct);
void OnNewOnlineProduct(class UOnlineProduct_TA* OnlineProduct, class FString Message);
class UOnlineProduct_TA* RemoveOnlineProductIdNoNotify(struct FProductInstanceID InstanceID);
void RemoveOnlineProductId(struct FProductInstanceID InstanceID);
bool GiveOnlineProductHelper(class UOnlineProduct_TA* OnlineProduct);
void GiveOnlineProduct(class UOnlineProduct_TA* OnlineProduct, class FString Message);
void GiveOnlineProductNoNotify(class UOnlineProduct_TA* OnlineProduct);
class UOnlineProduct_TA* GiveOnlineProductDataNoNotify(struct FOnlineProductData NewProductData);
class UOnlineProduct_TA* GiveOnlineProductData(struct FOnlineProductData NewProductData, class FString Message);
TArray<class UOnlineProduct_TA*> GiveOnlineProductDatas(TArray<struct FOnlineProductData> NewProductDatas);
TArray<class UProduct_TA*> GetNewProducts();
class UProduct_TA* GetProductFromOnlineID(struct FProductInstanceID OnlineProductId);
int32_t GetProductIdFromOnlineID(struct FProductInstanceID OnlineID);
TArray<class UOnlineProduct_TA*> GetOnlineProductsForSlot(class UProductSlot_TA* Slot);
bool HasAnyOnlineProductsForSlot(class UProductSlot_TA* Slot);
bool HasAnyOnlineProductOfID(int32_t ProductID);
class UOnlineProduct_TA* GetFirstOnlineProduct(int32_t ProductID);
class UOnlineProduct_TA* GetOnlineProduct(struct FProductInstanceID OnlineID);
struct FProductHashID GetNewestProductId();
bool IsProductOwned(struct FProductHashID HashID);
```

```

bool IsInstanceOwned(struct FProductInstanceID InstanceID);
void ValidateLoadoutsWithProduct(struct FProductInstanceID OnlineID);
void ValidateLoadouts();
void ValidateProfiles();
void RemoveAllProfiles();
class FString GetProfileName(int32_t LocalId);
class UProfile_TA* GetProfileForByLocalID(int32_t LocalId);
class UProfile_TA* GetProfileForPlayer(struct FUniqueNetId PlayerID);
void UpdateAllXEStatus();
void CacheLocalProducts();
void HandleOnlineGameInitialized();
void SetUncappedFramerate(unsigned long Value);
void SetUseLensFlares(unsigned long Value);
void SetUseVsync(unsigned long Value);
void HandleProfileSet(class ULocalPlayer_TA* Player);
void UploadSaveData();
void Save();
void HandlePostMapChange();
void HandleMapChange(class FString MapName);
void SetOnlineDirty(class USaveObject_TA* SaveObject);
void SetLocalDirty(class USaveObject_TA* SaveObject);
void HandleSaveObjectDirty(class USaveObject_TA* SaveObject);
void OnUnLoadSaveObject(class USaveObject_TA* SaveObject);
void OnLoadSaveObject(class USaveObject_TA* SaveObject);
void Unload();
void HandleContentConfigApplied(class UContentConfig_TA* ContentConfig);
void FixupLocalProducts();
void HandleDataLoaded(class USaveGameManager_TA* Manager, class UError* Error);
void eventConstruct();
void EventRedeemedOnlineProducts(class USaveData_TA* Data, TArray<class UOnlineProduct_TA*>& RedeemedOnlineProducts);
void EventSchematicResourcesChanged(class USaveData_TA* Data);
void EventOnlineSavedDataSynced(class USaveData_TA* Data);
void EventOnlineProductsSynced(class USaveData_TA* Data);
void EventRemovedOnlineProduct(class USaveData_TA* Data, class UOnlineProduct_TA* RemovedProduct);
void EventCrewRocketPassAcquired(class USaveData_TA* Data);
void EventNewOnlineProduct(class USaveData_TA* Data, class UOnlineProduct_TA* OnlineProduct, class FString Message);
void EventOnlineXPIinitialized(class USaveData_TA* Data);
};

// Class TAGame.GFxData_LocalPlayer_TA
// 0x00F0 (0x0098 - 0x0188)
class UGFxData_LocalPlayer_TA : public UGFxDataSingleton_X
{
public:
unsigned long          bOpenedStartMenu : 1;           // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bCheckedStartupEvents : 1;      // 0x0098 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long          bSaveDataLoaded : 1;           // 0x0098 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long          bBallcam : 1;                 // 0x0098 (0x0004)

```

```

[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bUseHoldBallCam : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long bUseBallCamIndicator : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000020] (CPF_EditInlineNotify)
unsigned long bAlternateNameplates : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000040] (CPF_EditInlineNotify)
unsigned long bMetric : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000080] (CPF_EditInlineNotify)
unsigned long bTeamColoredBoostMeter : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000100] (CPF_EditInlineNotify)
unsigned long bPrimaryPlayer : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000200] (CPF_EditInlineNotify)
unsigned long bPartyLeader : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000400] (CPF_EditInlineNotify)
unsigned long bIsInParty : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000800] (CPF_EditInlineNotify)
unsigned long bIsPlayerTrading : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00001000] (CPF_EditInlineNotify)
unsigned long bDemoBuild : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00002000] (CPF_EditInlineNotify)
unsigned long bForceSplitScreenUI : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00004000] (CPF_EditInlineNotify)
unsigned long bPrivateMatchGameOwner : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00008000] (CPF_EditInlineNotify)
unsigned long bAgeRestricted : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00010000] (CPF_EditInlineNotify)
unsigned long bPendingIdleKick : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00020000] (CPF_EditInlineNotify)
unsigned long bStartVoteToForfeitDisabled : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00040000] (CPF_EditInlineNotify)
unsigned long bDisableCrossPlay : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00080000] (CPF_EditInlineNotify)
unsigned long bReplacingBot : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00100000] (CPF_EditInlineNotify)
unsigned long bDisableDivisions : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00200000] (CPF_EditInlineNotify)
unsigned long bCheckedBootMessage : 1; // 0x0098 (0x0004)
[0x0000000000000000] [0x00400000]
unsigned long bUseSteamInput : 1; // 0x0098 (0x0004)
[0x0000004040000000] [0x00800000] (CPF_EditInlineNotify)
class FString SaveDataError; // 0x00A0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString OnlineError; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString PlayerName; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture* PlayerAvatarTexture; // 0x00D0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t XpLevel; // 0x00D8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString XPTitle; // 0x00E0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t XPTotal; // 0x00F0 (0x0004)

```

```

[0x0000000040000000] (CPF_EditInlineNotify)
int32_t XPPProgressInCurrentLevel; // 0x00F4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t XPRequiredForNextLevel; // 0x00F8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Team; // 0x00FC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float CameraYaw; // 0x0100 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t SplitScreenPosition; // 0x0104 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ControllerId; // 0x0108 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t LocalPlayerIndex; // 0x010C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FName LastMatchType; // 0x0110 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
float IdleKickSeconds; // 0x0118 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FProductHashID ContainerToOpenId; // 0x011C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t RankInfoDisplayType; // 0x0120 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t OrphanedControllerId; // 0x0121 (0x0001)
[0x0000000000002000] (CPF_Transient)
class UPersonaSave_TA* PersonaSave; // 0x0128 (0x0008)
[0x0000000000000000]
struct FUniqueNetId PlayerID; // 0x0130 (0x0048)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class ULocalPlayer_TA* Player; // 0x0178 (0x0008)
[0x0000004000002000] (CPF_Transient)
class UGFxModal_X* ModalOrphaned; // 0x0180 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_LocalPlayer_TA");
}

return uClassPointer;
};

bool CanShowRankInfoForPlaylist(class UGameSettingPlaylist_X* InPlaylist);
bool CanShowRankInfoForPlaylistByID(int32_t InPlaylistID);
bool CanShowRankInfoForPlaylistByName(class FString InPlaylistName);
void HandlePersonaSaveLoaded(class UPersonaSave_TA* InPersonaSave);
void HandleXPUloaded(class UClientXPSave_TA* XPSave);
void HandleXPLoaded(class UClientXPSave_TA* XPSave);
void HandleBallCamModeChanged();

```

```
void HandleCameraSettingsLoaded(class UProfileCameraSave_TA* CameraSettings);
void HandleProfileGameplaySettingsLoaded(class UProfileGameplaySave_TA* GameplaySettings);
void ClearContainerToOpen();
bool HasBlueprintToPreview();
bool HasContainerToOpen();
void HandleCrossPlatformEnabledChanged();
void HandleAccountSettingsAdded(class UAccountSettingsComponent_TA* AccountSettings);
void OnExitToStartScreen(class UGFxModal_X* Modal);
void OnPickAccount(class UGFxModal_X* Modal);
void ShowOrphanedUserDialog(uint8_t InControllerID);
void ShowSecondaryPlayerOrphanedModal();
void ShowPrimaryPlayerOrphanedModal();
void OnUserRestored(uint8_t InControllerID);
void OnUserOrphaned(uint8_t InControllerID);
void HandleVoteToForfeitDisabledChanged(class APRI_TA* PRI);
void HandleCanPlayOnlineChanged(class UOnlinePlayer_X* P);
void HandleLocalPlayerChanged(class ULocalPlayer* NewPlayer);
void ClearLocalPlayerData(class AGameEvent_TA* GameEvent);
void HandlePrivateMatchGameOwnerUpdated(class AGameEvent_TA* GameEvent, class APRI_TA* NewGameOwner);
void ChangeName(class FString NewName);
void Spectate();
void OnJoinTeamFailed(int32_t TeamNum);
void OnServerChangeTeamFailed(int32_t TeamNum);
void ChangeTeam(int32_t TeamIndex);
class APRI_TA* GetPRI();
bool GetOverlayEnabled();
void SetCheckedStartupEvents(unsigned long bHasChecked);
void SetOpenedStartMenu(unsigned long bHasBeenOpened);
void UpdateXP(class UClientXPSave_TA* XPSave);
void OnXpInitialized(int32_t NewXP);
void HandleSplitScreenPositionChanged(class ULocalPlayer_TA* LocalPlayer);
class UOnlineGame_X* GetOnlineGame();
void HandleGetAvatar(struct FUniqueNetId InPlayerId, class UTexture* Avatar, class FString OnlinePlayerName);
void HandleReceivedPRI(class APlayerController_X* PC);
void HandleReceivedController(class UPlayer* PlayerRef);
void CheckBootMessages();
void HandleOnlineNameChanged(class UOnlinePlayer_X* P);
void HandlePsyNetConnected(class UPsyNetConnection_X* C);
void HandleLoginStatusChanged(class UOnlinePlayer_X* OnlinePlayer);
void HandleSyncSuccess(class UOnlineStorageSyncManager_TA* OnlineStorageSyncManager);
void HandleGamepadInputAPIChanged(class UGameViewportClient* InGVC, int32_t InControllerID, uint8_t InInputAPI);
void HandleProfileSet(class ULocalPlayer_TA* InLocalPlayer);
void HandleSplitScreenTypeChanged(class UGameViewportClient_TA* GVC);
void HandleDrawWorldFullScreenChanged(class UGameViewportClient_TA* GVC);
void UpdateName();
void UpdatePrimaryPlayer();
void HandlePrimaryPlayerChange(class ULocalPlayer* OldPrimary, class ULocalPlayer* NewPrimary);
void eventOnRemoved();
void eventOnShellSet();
```

```

};

// Class TAGame.__GFxData_MenuTree_TA__EnterTreeAtRoot_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_MenuTree_TA__EnterTreeAtRoot_0x1 : public UObject
{
public:
    struct FName InRootID; // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_MenuTree_TA__EnterTreeAtRoot_0x1");
        }

        return uClassPointer;
    };

    bool __GFxDATA_MenuTree_TA__EnterTreeAtRoot_0x1(class UMenuTreeNode_TA* N);
};

// Class TAGame.MenuTreeNode_TA
// 0x0118 (0x0060 - 0x0178)
class UMenuTreeNode_TA : public UObject
{
public:
    struct FName NodeID; // 0x0060 (0x0008)
    [0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
    class FString Title; // 0x0068 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString Description; // 0x0078 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class UClass* GFxDATAClass; // 0x0088 (0x0008)
    [0x0001008000000000]
    class UGFxDATA_MenuTreeNode_TA* GFxNode; // 0x0090
    (0x0008) [0x0001008000000000]
    class UMenuTreeBranch_TA* ParentNode; // 0x0098 (0x0008)
    [0x0001000000000000]
    class UTexture* ThumbnailImage; // 0x00A0 (0x0008)
    [0x0001000000000001] (CPF_Edit)
    class UMenuTreeNodeCrumbTrail_TA* CrumbTrail; // 0x00A8
    (0x0008) [0x0001000000000000]
    class FString LocalizationCategory; // 0x00B0 (0x0010)
    [0x0001004000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString TitleKey; // 0x00C0 (0x0010)
    [0x0001004000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString DescriptionKey; // 0x00D0 (0x0010)
    [0x0001004000400001] (CPF_Edit | CPF_NeedCtorLink)

```

```

class FString SecondaryTitle; // 0x00E0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t SecondaryTitleTimeAmount; // 0x00F0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
unsigned long bUppercaseSecondaryTitle : 1; // 0x00F4 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
class FString BadgeText; // 0x00F8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
uint8_t BadgeType; // 0x0108 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
TArray<uint8_t> EnabledRequirements; // 0x0110 (0x0010)
[0x0001000004000001] (CPF>Edit | CPF_NeedCtorLink)
TArray<uint8_t> SupportedEngagementEvents; // 0x0120
(0x0010) [0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FNodePlatformAvailability PlatformAvailability; // 0x0130
(0x0018) [0x0001004000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FScriptDelegate __EventNodeClicked__Delegate; // 0x0148
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventNodeFirstTimeClicked__Delegate; // 0x0160
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_TA");
}

return uClassPointer;
};

bool __MenuTreeNode_TA__OnClicked_0x1(struct FMenuTreeFocusMap N);
bool __MenuTreeNode_TA__IsAvailableOnCurrentPlatform_0x1(uint8_t Platform);
void HandleCrumbTrailUpdated();
bool IsDescendentOf(struct FName InAncestorID);
bool IsAvailableOnCurrentPlatform();
bool IsVisible();
TArray<class UMenuTreeNode_TA*> GetActiveNodes();
void OnClicked();
void Cleanup();
void OnGFxNodeSet();
void SetGFxNode(class UGFxData_MenuTreeNode_TA* InGFxNode);
void OnInit();
void EventNodeFirstTimeClicked(struct FName InNodeID);
void EventNodeClicked(struct FName InNodeID);
};

// Class TAGame.GFxData_MenuTree_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_MenuTree_TA : public UGFxDataSingleton_X
{

```

```
public:  
class UMenuTreeConfig_TA*           MenuTreeConfig;          // 0x0098  
(0x0008) [0x0001800000000000]  
class UMenuTreeBuilder_TA*          MenuTreeBuilder;        // 0x00A0  
(0x0008) [0x0001800000000000]  
class UMenuTreeBranch_TA*           rootNode;             // 0x00A8 (0x0008)  
[0x0001000000000000]  
class UMenuTreeBranch_TA*           currentNode;           // 0x00B0 (0x0008)  
[0x0001004000000000]  
TArray<struct FMenuTreeNodeData>    NodesData;              // 0x00B8  
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MenuTree_TA");  
}  
  
return uClassPointer;  
};  
  
void __GFxData_MenuTree_TA__OnShellSet_0x1(class FString _);  
bool __GFxData_MenuTree_TA__GetFocusedChildIndex_0x1(struct FMenuTreeFocusMap N);  
void __GFxData_MenuTree_TA__Cleanup_0x1(class UMenuTreeNode_TA* N);  
void StartMatchmaking();  
class FString GetLastPlaylistFolderTitle();  
bool IsRankedView();  
void Cleanup();  
int32_t GetFocusedChildIndex();  
void SetCurrentNode(class UMenuTreeBranch_TA* NewCurrentNode);  
int32_t GetNodeDataIndex(struct FName NodeID);  
int32_t GetNodeRowIndex(struct FName NodeID);  
void OnBackClicked();  
void OnChildNodeClicked(struct FName ChildID);  
void OnFirstTimeClicked(struct FName NodeID);  
void ExitTree();  
void JumpToParentBranch(struct FName InChildID);  
void JumpToBranch(struct FName InBranchID);  
void EnterTreeAtRoot(struct FName InRootID);  
void eventOnShellSet();  
};  
  
// Class TAGame.__GFxData_MenuTree_TA__GetFocusedChildIndex_0x2  
// 0x000C (0x0060 - 0x006C)  
class U_GFxData_MenuTree_TA__GetFocusedChildIndex_0x2 : public UObject  
{  
public:  
class UPlayMenuSave_TA*            PlayMenuSave;           // 0x0060 (0x0008)  
[0x0001000000000000]
```

```

int32_t FocusNodeMappingIndex; // 0x0068 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_MenuTree_TA__GetFocusedChildIndex_0x2");

}

return uClassPointer;
};

bool __GFxDATA_MenuTree_TA__GetFocusedChildIndex_0x2(struct FMenuTreeNodeData
N);
};

// Class TAGame.PlayMenuSave_TA
// 0x0030 (0x00C8 - 0x00F8)
class UPlayMenuSave_TA : public USaveObject_TA
{
public:
unsigned long bUsingMultiselect : 1; // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000001]
struct FName LastVisitedPlaylistFolderID; // 0x00CC (0x0008)
[0x0000000000000000]
TArray<struct FMenuTreeFocusMap> FocusSave; // 0x00D8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> FirstTimeClickedPlaylistIDs; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayMenuSave_TA");
}

return uClassPointer;
};

};

// Class TAGame._GFxDATA_MenuTree_TA__GetLastPlaylistFolderTitle_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_MenuTree_TA__GetLastPlaylistFolderTitle_0x1 : public UObject
{

```

```
public:  
class UPlayMenuSave_TA*           PlayMenuSave;          // 0x0060 (0x0008)  
[0x0001000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__GFxData_MenuTree_TA__GetLastPlaylistFolderTitle_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __GFxData_MenuTree_TA__GetLastPlaylistFolderTitle_0x1(class UMenuTreeNode_TA* N);  
};  
  
// Class TAGame.__GFxData_MenuTree_TA__JumpToBranch_0x1  
// 0x0008 (0x0060 - 0x0068)  
class U__GFxData_MenuTree_TA__JumpToBranch_0x1 : public UObject  
{  
public:  
struct FName                  InBranchID;           // 0x0060 (0x0008)  
[0x0001000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__GFxData_MenuTree_TA__JumpToBranch_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __GFxData_MenuTree_TA__JumpToBranch_0x1(class UMenuTreeNode_TA* N);  
};  
  
// Class TAGame.__GFxData_MenuTree_TA__JumpToParentBranch_0x1  
// 0x0008 (0x0060 - 0x0068)  
class U__GFxData_MenuTree_TA__JumpToParentBranch_0x1 : public UObject  
{  
public:  
struct FName                 InChildID;            // 0x0060 (0x0008)  
[0x0001000000000000]
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_MenuTree_TA__JumpToParentBranch_0x1");
}

return uClassPointer;
};

bool __GFxData_MenuTree_TA__JumpToParentBranch_0x1(class UMenuTreeNode_TA* N);
};

// Class TAGame.GFxData_MTXGarage_TA
// 0x00F8 (0x0098 - 0x0190)
class UGFxData_MTXGarage_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FMTXPurchaseInfo> CatalogItems; // 0x0098
(0x0010) [0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FMCatalogInfo> CatalogProducts; // 0x00A8
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCartInfo> CartItems; // 0x00B8 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString CartTotal; // 0x00C8 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString Currency; // 0x00D8 (0x0010)
[0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long TransactionInProgress : 1; // 0x00E8 (0x0004)
[0x000000000002000] [0x00000001] (CPF_Transient)
unsigned long bCatalogReady : 1; // 0x00E8 (0x0004)
[0x0000000000000000] [0x00000002]
class UGFxModal_X* ModalProcessing; // 0x00F0 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t BlackMarketSkinProductID; // 0x00F8 (0x0004)
[0x0000000000000000]
float CatalogExpirationTime; // 0x00FC (0x0004)
[0x0000000000002000] (CPF_Transient)
float CatalogCacheDuration; // 0x0100 (0x0004)
[0x0000000000000000]
uint8_t CatalogCategory; // 0x0104 (0x0001)
[0x0000000000002000] (CPF_Transient)
float PurchaseTimeoutTime; // 0x0108 (0x0004)
[0x0000000000000002] (CPF_Const)
float GetPriceTimeoutTime; // 0x010C (0x0004)
[0x0000000000000002] (CPF_Const)
class AGameInfo_GFxMenu_TA* Menu; // 0x0110 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UMtxConfig_TA* MtxConfig; // 0x0118 (0x0008)
[0x0000800000000001] (CPF>Edit)

```

```

class UOnlineSystemInterface* SystemInterface_Object; // 0x0120
(0x0008) [0x0000000000000000]
class UOnlineSystemInterface* SystemInterface_Interface; // 0x0128
(0x0008) [0x0000000000000000]
class UOnlinePurchaseInterface* PurchaseInterface_Object; // 0x0130
(0x0008) [0x0000000000000000]
class UOnlinePurchaseInterface* PurchaseInterface_Interface; // 0x0138
(0x0008) [0x0000000000000000]
class UPsyNetConnection_X* PsyNetConnection; // 0x0140
(0x0008) [0x0000000000000000]

class FString RegionRestrictedOpenCrate; // 0x0148 (0x0010)
[0x00100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UOnlineProductStoreSet_TA* OnlineProductStoreSet; // 0x0158
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
struct FScriptDelegate __EventCatalogReady__Delegate; // 0x0160
(0x0018) [0x00000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventMTXPurchaseSuccess__Delegate; // 0x0178
(0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MTXGarage_TA");
}

return uClassPointer;
};

struct FProductHashID __GFxData_MTXGarage_TA__GetKeyHashIDs_0x2(class
UOnlineProduct_TA* Op);
bool __GFxData_MTXGarage_TA__GetKeyHashIDs_0x1(class UOnlineProduct_TA* Op);
struct FProductHashID __GFxData_MTXGarage_TA__GetDecryptorHashIDs_0x2(class
UOnlineProduct_TA* Op);
bool __GFxData_MTXGarage_TA__GetDecryptorHashIDs_0x1(class UOnlineProduct_TA* Op);
struct FProductHashID __GFxData_MTXGarage_TA__GetTradeHoldSortedKeys_0x2(class
UOnlineProduct_TA* Op);
bool __GFxData_MTXGarage_TA__GetTradeHoldSortedKeys_0x1(class UOnlineProduct_TA* Op);
bool __GFxData_MTXGarage_TA__AnyKeyRestricted_0x1(class UOnlineProduct_TA* Key);
struct FProductInstanceId __GFxData_MTXGarage_TA__UnlockContainers_0x2(class
UOnlineProduct_TA* Op);
struct FProductInstanceId __GFxData_MTXGarage_TA__UnlockContainers_0x1(class
UOnlineProduct_TA* Op);
int32_t __GFxData_MTXGarage_TA__RevealContainerUnlocks_0x4(class UOnlineProduct_TA* P);
struct FProductHashID __GFxData_MTXGarage_TA__RevealContainerUnlocks_0x3(class
UOnlineProduct_TA* P);
class FString __GFxData_MTXGarage_TA__HandleGetCatalogResponse_0x2(struct FCurrency C);
struct FMTCartItem __GFxData_MTXGarage_TA__CheckoutAndReturnError_0x2(struct FCartInfo
Item);
bool __GFxData_MTXGarage_TA__CheckoutAndReturnError_0x1(struct FCartInfo Item);

```

```
void HandleMTXErrorType(class UErrorType* InErrorType);
void HandleMTXError(class UError* Error);
void OnExitDisplay();
void SetDisplayProduct(struct FProductHashID HashID);
void ProductRevealExit();
void OnRevealFinished(int32_t SlotIndex);
void ProductReveal();
void OnKeyPurchase(TArray<struct FProductInstanceID> InstanceID);
void HandlePsynetConfirmPurchase(class URPC_MicroTransactions_FinalizePurchase_TA*
RPC);
void OnPlayerConfirmPurchase(unsigned long bAuthorized, uint64_t OrderId);
void RefreshCurrentCatalog();
void HandleClaimEntitlements(class UOnlinePlayerMTX_TA* MTX, TArray<class
UOnlineProduct_TA*> NewProducts);
void HandleStartPurchase(class URPC_MicroTransactions_StartPurchase_TA* RPC);
void MTXPurchaseEnd();
void PurchaseTimeout();
class UErrorType* IsUserAbleToPurchaseReturnError();
bool IsUserAbleToPurchaseNoModal();
bool IsUserAbleToPurchase();
class UErrorType* CheckoutAndReturnError();
class UErrorType* CheckoutAndReturnError_V1();
void CheckOut();
void CheckoutCompleteFlashNNX();
void CheckoutCompleteNNX();
TArray<struct FMTCartItem> MapCartToCatalog();
void UpdateCartTotal();
void AddToCart(int32_t CatalogID, int32_t Count);
void RemoveFromCart(int32_t CatalogID, int32_t Count);
void ClearCart();
void OnGetCatalog(unsigned long bSuccess);
void HandleGetPriceInfoComplete(struct FName ProductName, class FString Price, class FString
DiscountPrice, int32_t DiscountPercentage);
void HandleGetPriceTimeout();
void UpdatePriceInfo(TArray<struct FMCatalogInfo> PriceRequests);
void HandleImageDownloaded(class FString URL, class UTexture2DDynamic* Texture);
void HandleGetCatalogResponse(class URPC_MicroTransactions_GetCatalog_TA* RPC);
void SetCatalogCategory(uint8_t InCategory);
void GetCatalog();
void OnUnlockFailed();
void OnUnlock(TArray<struct FProductHashID> HashIDs, TArray<int32_t> ProductIDs);
void RevealContainerUnlocks(unsigned long bMultiCrateUnlock, TArray<struct
FOnlineProductData>& Drops);
void HandleUnlockContainersResponse(class URPC_MicroTransactions_UnlockContainer_TA*
RPC);
void UnlockContainerFailed();
void UnlockContainers(struct FProductHashID ContainerHashID, TArray<struct FProductHashID>
KeyHashIDs);
bool AnyKeyRestricted(TArray<class UOnlineProduct_TA*>& KeyProducts);
TArray<struct FProductHashID> GetTradeHoldSortedKeys(unsigned long bIsAscending);
TArray<struct FProductHashID> GetDecryptorHashIDs();
TArray<struct FProductHashID> GetKeyHashIDs();
void OnKeyListUpdated();
bool CanStartTransaction();
```

```

void eventOnRemoved();
void eventOnShellSet();
void EventMTXPurchaseSuccess(TArray<struct FOnlineProductData>& Products);
void EventCatalogReady(struct FMTXPurchaseInfo Info, uint8_t Category);
};

// Class TAGame.__GFxData_MTXGarage_TA__HandleGetCatalogResponse_0x1
// 0x00A0 (0x0060 - 0x0100)
class U_GFxDATA_MTXGarage_TA_HandleGetCatalogResponse_0x1 : public UObject
{
public:
    struct FMTXPurchaseInfo           BundleInfo;          // 0x0060 (0x00A0)
    [0x000000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_MTXGarage_TA_HandleGetCatalogResponse_0x1");
        }

        return uClassPointer;
    };

    void __GFxDATA_MTXGarage_TA_HandleGetCatalogResponse_0x1(class UTexture2DDynamic*
Texture);
};

// Class TAGame.__GFxDATA_MTXGarage_TA_HandleUnlockContainersResponse_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_MTXGarage_TA_HandleUnlockContainersResponse_0x1 : public UObject
{
public:
    class USaveData_TA*               PCSaveData;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_MTXGarage_TA_HandleUnlockContainersResponse_0x1");
        }

        return uClassPointer;
    };
}

```

```

void __GFxData_MTXGarage_TA_HandleUnlockContainersResponse_0x2(struct
FProductInstanceID KeyInstanceID);
void __GFxData_MTXGarage_TA_HandleUnlockContainersResponse_0x1(struct
FProductInstanceID InstanceID);
};

// Class TAGame.__GFxData_MTXGarage_TA_RevealContainerUnlocks_0x1
// 0x0020 (0x0060 - 0x0080)
class U__GFxData_MTXGarage_TA_RevealContainerUnlocks_0x1 : public UObject
{
public:
TArray<class UOnlineProduct_TA*> OnlineProducts; // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UProductsSave_TA* ProductsSave; // 0x0070 (0x0008)
[0x0000000000000000]
class USaveData_TA* PCSaveData; // 0x0078 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_MTXGarage_TA_RevealContainerUnlocks_0x1");
}

return uClassPointer;
};

void __GFxData_MTXGarage_TA_RevealContainerUnlocks_0x2(class UOnlineProduct_TA* Op);
void __GFxData_MTXGarage_TA_RevealContainerUnlocks_0x1(struct FOnlineProductData D);
};

// Class TAGame.ProductsSave_TA
// 0x00A0 (0x00C8 - 0x0168)
class UProductsSave_TA : public USaveObject_TA
{
public:
TArray<uint64_t> OnlineProductInstanceIDs; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<uint64_t> NewProductIDs; // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<uint64_t> UnseenContainerProducts; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
uint64_t LastUnlockDisplayId; // 0x00F8 (0x0008)
[0x0000000000000000]
TArray<struct FProductInstanceID> OnlineProductInstanceIDs128; // 0x0100
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceID> NewProductIDs128; // 0x0110
(0x0010) [0x0008000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceID> UnseenContainerProducts128; // 0x0120

```

```

(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
struct FProductInstanceID           LastUnlockDisplayId128;          // 0x0130
(0x0010) [0x0000000000000000]
int32_t                           PreviousKeyCount;                // 0x0140 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                           PreviousDecryptorCount;        // 0x0144 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
unsigned long                      bHasUnseenGiftboxes : 1;       // 0x0148 (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
struct FScriptDelegate             __NewProductIDs128__ChangeNotify; // 0x0150
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductsSave_TA");
}

return uClassPointer;
};

bool __ProductsSave_TA__OnLoad_0x1(struct FProductInstanceID InstanceID);
bool __ProductsSave_TA__GetUnseenContainerDrops_0x1(struct FProductHashID ProductHash);
void __ProductsSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __NewProductIDs128__ChangeNotifyFunc();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
TArray<struct FProductHashID> GetUnseenContainerDrops();
void HandleOnlineProductsChanged(TArray<class UOnlineProduct_TA*> InOnlineProducts);
void RemoveContainerDrop(struct FProductHashID HashID);
void ClearUnseenContainerDrops();
struct FProductHashID GetHighestRarityDrop();
bool IsUnseenContainerDrop(struct FProductInstanceID InstanceID);
void AddUnseenContainerDrop(struct FProductInstanceID InstanceID);
void OnUnseenGiftboxesUpdated();
bool ProductIsNew(struct FProductInstanceID OnlineID);
void UnlockProduct(int32_t ProductID);
void LockProduct(int32_t ProductID);
void ClearNewProducts();
void OnLoad();
};

// Class TAGame.__GFxDATA_MultiItemDrops_TA__DropItemGroup_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_MultiItemDrops_TA__DropItemGroup_0x1 : public UObject
{
public:
class UItemDropGroup_TA*           DropGroup;                  // 0x0060 (0x0008)
[0x0000000000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_MultiltemDrops_TA__DropItemGroup_0x1");
}

return uClassPointer;
};

void __GFxData_MultiltemDrops_TA__DropItemGroup_0x1(struct FOnlineProductData
OnlineProduct);
};

// Class TAGame.GFxData_MultiltemDrops_TA
// 0x0058 (0x0098 - 0x00F0)
class UGFxData_MultiltemDrops_TA : public UGFxDataSingleton_X
{
public:
class UMultiltemDropConfig_TA* MultiltemDropConfig; // 0x0098
(0x0008) [0x0000800000000000]
class UGFxData_ItemDropGroup_TA* ActiveGroup; // 0x00A0
(0x0008) [0x0000000000000000]
TArray<class UltimDropGroup_TA*> PendingGroups; // 0x00A8
(0x0010) [0x00000000400000] (CPF_NeedCtorLink)
int32_t GroupsRemaining; // 0x00B8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString DropTableName; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UWallet_TA* Wallet; // 0x00D0 (0x0008)
[0x0000000000000000]
TArray<struct FOnlineXPReward> RewardDrops; // 0x00D8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FName DropGroupType_ALL; // 0x00E8 (0x0008)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MultiltemDrops_TA");
}

return uClassPointer;
};

void __GFxData_MultiltemDrops_TA__OnShellSet_0x3(class UWallet_TA* InWallet);

```

```
void __GFxDATA_MultiItemDrops_TA__OnShellSet_0x5(class UWallet_TA* W, class FString Message, struct FCurrency& Drop);
void __GFxDATA_MultiItemDrops_TA__OnShellSet_0x4(class UWallet_TA* W, class FString Message, struct FCurrency& Drop);
void __GFxDATA_MultiItemDrops_TA__OnShellSet_0x2(class UShopsManager_TA* InShopsManager);
void __GFxDATA_MultiItemDrops_TA__OnShellSet_0x1(class APlayerControllerBase_TA* InController);
bool IsProductInGroup(struct FProductInstanceID InstanceID);
TArray<struct FProductHashID> GetUnseenContainerDrops();
void RemoveContainerDrop(struct FProductHashID HashID);
void ClearContainerDrops();
void OnAllItemDropsHandled();
bool AnyPlayerHasContainerDrops();
bool AnyPlayerHasUnlock();
void PlayerAllDropsHandled();
class UProfile_TA* GetProfile();
void EquipNow(struct FProductHashID HashID);
bool CanEquipNow(struct FProductHashID HashID);
void AllGroupsShown();
void ShowedGroup();
void ShowNextDropType(struct FName InTableName);
int32_t GetPendingDropCount(struct FName InTableName);
void HandleGroupSeen(class UItemDropGroup_TA* DropGroup);
void HandleActiveGroupSeen();
void RemoveInvalidDrops();
void ClearDropsByType(struct FName DropType);
TArray<class UItemDropGroup_TA*> GetDropsByType(struct FName DropType);
void RemoveDropsOfClassType(class UClass* DropClass);
void ClearActiveGroup();
void HandleChallengeRewardCollected(class UChallengeManager_TA* ChallengeManager, class UChallenge_TA* Challenge, TArray<class UOnlineProduct_TA*> OnlineProductRewards, struct FOnlineReward& Reward);
void HandleReceivedXPDrop(class URewardDrop_TA* RewardDrop);
class UCompleteTaskDropGroup_TA* DropRewardGroup(class UClass* CompleteTaskClass, struct FOnlineReward Reward, class FString Title, class UTexture* Icon, int32_t XPChange, int32_t PointChange, TArray<struct FCurrency>& Currency);
void AddProductToGroup(struct FProductInstanceID InstanceID, uint64_t AddedTimestamp, class UItemDropGroup_TA* ItemGroup);
void AddOnlineProductToGroup(class UItemDropGroup_TA* ItemGroup, struct FOnlineProductData& ProductData);
void HandleNewOnlineItem(class USaveData_TA* InSaveData, class UOnlineProduct_TA* OnlineProduct, class FString Message);
void VerifyHashIDs(class UItemDropGroup_TA* InDropGroup);
void HandleCrewRocketPassAcquired(class USaveData_TA* Data);
void HandleOnlineProductsSynced(class USaveData_TA* Data);
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA* SaveData, class UError* Error);
void MakeGroupActive(class UItemDropGroup_TA* InActiveGroup);
void MakeFirstPendingGroupActive();
void MakeGroupActiveOnFTECompleted();
void OnGroupAdded();
void RequeueActiveGroup();
void CheckAndSetActiveGroup();
```

```

void DropCrewRocketPassRefund(struct FCurrency Drop, class FString Message);
void DropCrewRocketPass();
class UItemDropGroup_TA* InternalAddGroup(class UItemDropGroup_TA* Group);
class UItemDropGroup_TA* AddGroup(class UClass* GroupClass, uint8_t DisplayOrder, class
FString Message);
void DropXPGroup();
void DropItemGroup(class FString Message, TArray<struct FOnlineProductData>&
OnlineProducts, TArray<struct FCurrency>& CurrencyDrops);
struct FCurrency CreateCurrencyDrop(int32_t Id, int32_t Amount);
void ProcessCurrencyDrop(struct FCurrency& Drop);
uint8_t GetDisplayOrder(class UProduct_TA* Product);
void HandleShopPurchase(TArray<struct FOnlineProductData> PurchasedProducts,
TArray<struct FCurrency> PurchasedCurrencies);
void HandleReceivedCurrencyDrop(class APlayerControllerBase_TA* PC, struct FCurrency Drop);
void eventOnShellSet();
};

// Class TAGame.__GFxData_MultiltemDrops_TA__DropRewardGroup_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_MultiltemDrops_TA__DropRewardGroup_0x1 : public UObject
{
public:
class UCompleteTaskDropGroup_TA*           DropGroup;          // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_MultiltemDrops_TA__DropRewardGroup_0x1");
}

return uClassPointer;
};

void __GFxData_MultiltemDrops_TA__DropRewardGroup_0x1(struct FOnlineProductData P);
};

// Class TAGame.CompleteTaskDropGroup_TA
// 0x001C (0x00C0 - 0x00DC)
class UCompleteTaskDropGroup_TA : public UItemDropGroup_TA
{
public:
class UTexture*                           Icon;              // 0x00C0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString                            Title;             // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                                  Points;            // 0x00D8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CompleteTaskDropGroup_TA");
}

return uClassPointer;
};

};

// Class TAGame.__GFxData_MultiltemDrops_TA__GetDropsByType_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_MultiltemDrops_TA__GetDropsByType_0x1 : public UObject
{
public:
struct FName DropType; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_MultiltemDrops_TA__GetDropsByType_0x1");
}

return uClassPointer;
};

bool __GFxData_MultiltemDrops_TA__GetDropsByType_0x1(class UItemDropGroup_TA* Group);
};

// Class TAGame.__GFxData_MultiltemDrops_TA__HandleChallengeRewardCollected_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxData_MultiltemDrops_TA__HandleChallengeRewardCollected_0x1 : public UObject
{
public:
TArray<struct FCurrency> Currency; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_MultiltemDrops_TA__HandleChallengeRewardCollected_0x1");
}

return uClassPointer;
};

void __GFxData_MultiltemDrops_TA__HandleChallengeRewardCollected_0x1(class
UChallengeCurrency_TA* C);
};

// Class TAGame.__GFxData_MultiltemDrops_TA__HandleGroupSeen_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_MultiltemDrops_TA__HandleGroupSeen_0x1 : public UObject
{
public:
class USaveData_TA*           SaveData;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_MultiltemDrops_TA__HandleGroupSeen_0x1");
}

return uClassPointer;
};

void __GFxData_MultiltemDrops_TA__HandleGroupSeen_0x1(struct FProductDrop P);
};

// Class TAGame.__GFxData_MultiltemDrops_TA__HandleNewOnlineItem_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxData_MultiltemDrops_TA__HandleNewOnlineItem_0x1 : public UObject
{
public:
class USaveData_TA*           InSaveData;        // 0x0060 (0x0008)
[0x0000000000000000]
class UOnlineProduct_TA*       OnlineProduct;    // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class
TAGame._GFxData_MultiltemDrops_TA__HandleNewOnlineItem_0x1");
}

return uClassPointer;
};

void __GFxData_MultiltemDrops_TA__HandleNewOnlineItem_0x1(class UPlayer* P);
};

// Class TAGame._GFxData_MultiltemDrops_TA__IsProductInGroup_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxData_MultiltemDrops_TA__IsProductInGroup_0x1 : public UObject
{
public:
struct FProductInstanceId           InstanceID;                      // 0x0060 (0x0010)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_MultiltemDrops_TA__IsProductInGroup_0x1");
}

return uClassPointer;
};

bool
__GFxData_MultiltemDrops_TA__IsProductInGroup_0x1__GFxData_MultiltemDrops_TA__IsPro
ductInGroup_0x2_0x1(struct FProductDrop Product);
bool __GFxData_MultiltemDrops_TA__IsProductInGroup_0x2(class UItemDropGroup_TA* Group);
bool __GFxData_MultiltemDrops_TA__IsProductInGroup_0x3(struct FProductDrop Product);
bool __GFxData_MultiltemDrops_TA__IsProductInGroup_0x1(struct FProductDrop Product);
};

// Class TAGame._GFxData_MusicPlayer_TA__GetTitleTrackBuilder_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxData_MusicPlayer_TA__GetTitleTrackBuilder_0x1 : public UObject
{
public:
class UIMusicTracklistBuilder*      PreviousBuilder_Object;        // 0x0060
(0x0008) [0x0000000000000000]
class UIMusicTracklistBuilder*      PreviousBuilder_Interface;    // 0x0068
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_MusicPlayer_TA__GetTitleTrackBuilder_0x1");
}

return uClassPointer;
};

void
__GFxData_MusicPlayer_TA__GetTitleTrackBuilder_0x1__GFxData_MusicPlayer_TA__GetTitleT
rackBuilder_0x1_0x1();
void __GFxData_MusicPlayer_TA__GetTitleTrackBuilder_0x1(class UMusicTrack_TA* _);
};

// Class TAGame.IMusicTracklistBuilder
// 0x0000 (0x0060 - 0x0060)
class UIMusicTracklistBuilder : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IMusicTracklistBuilder");
}

return uClassPointer;
};

class UMusicTracklist_TA* CreateMusicTracklist(TArray<class UMusicTrack_TA*> Tracks);
};

// Class TAGame.MusicTrack_TA
// 0x0028 (0x0060 - 0x0088)
class UMusicTrack_TA : public UObject
{
public:
class UAkSoundCue*           Cue;           // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UTexture*               Icon;          // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FScriptDelegate         __NextTrackAction__Delegate; // 0x0070
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTrack_TA");
}

return uClassPointer;
};

void NextTrackAction(class UMusicTrack_TA* Track);
};

// Class TAGame.GFxData_MusicPlayer_TA
// 0x0110 (0x0098 - 0x01A8)
class UGFxData_MusicPlayer_TA : public UGFxDataSingleton_X
{
public:
class UMusicUrlConfig_TA*           UrlConfig;          // 0x0098 (0x0008)
[0x0001000000000001] (CPF_Edit)
unsigned long                      bDebugMusicPlayer : 1; // 0x00A0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long                      bPlaylistsEnabled : 1; // 0x00A0 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long                      bEnableMusicControls : 1; // 0x00A0 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long                      bPlayerPopupDisplaying : 1; // 0x00A0 (0x0004)
[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long                      bActivePlaylistsEnabled : 1; // 0x00A0 (0x0004)
[0x0000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long                      bTitleTracksActive : 1; // 0x00A0 (0x0004)
[0x0000000000002000] [0x00000020] (CPF_Transient)
TArray<struct FGFxMusicPlaylist>   DefaultPlaylists;    // 0x00A8 (0x0010)
[0x0000000004000000] (CPF_NeedCtorLink)
TArray<struct FGFxMusicPlaylist>   Playlists;          // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString                     TrackTitle;         // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString                     TrackArtist;        // 0x00D8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UTexture*                  TrackIcon;          // 0x00E8 (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString                     TrackUrl;          // 0x00F0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
float                           StartupLogoDisplayDurationMS; // 0x0100 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString                     CurrentTrackName;    // 0x0108 (0x0010)
[0x0008000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                         CurrentTrackPlayingID; // 0x0118 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UMusicTrack_TA*            CurrentTrack;        // 0x0120 (0x0008)
[0x0000000000002000] (CPF_Transient)
TArray<struct FPlaylistTrack>   MasterTracklist;     // 0x0128 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UIMusicTracklist_TA*      ActiveTracklist_Object; // 0x0138 (0x0008)

```

```

[0x0000000000000000]
class UIMusicTracklist_TA*           ActiveTracklist_Interface;          // 0x0140
(0x0008) [0x0000000000000000]
class UMusicTracklistFactory_TA*     TracklistFactory;                  // 0x0148
(0x0008) [0x0000000000000000]
class UMusicTracklistBuildStep_RemoveTrack_TA* MainRotationTracklistBuilder;    //
0x0150 (0x0008) [0x0000000000000000]
class UMusicTheme_TA*                MusicTheme;                      // 0x0158 (0x0008)
[0x000000000002000] (CPF_Transient)
class UMusicPlayerSave_TA*          MusicSave;                       // 0x0160 (0x0008)
[0x000000000002000] (CPF_Transient)
class UGameplayMusicPlayer_TA*       GameMusicPlayer;                 // 0x0168
(0x0008) [0x000000000002000] (CPF_Transient)
class UStreamerSafeConfig_TA*       StreamerSafeConfig;              // 0x0170
(0x0008) [0x0001800000000000]
struct FScriptDelegate             __EventTrackFinished__Delegate;      // 0x0178
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __CurrentTrackName__ChangeNotify;   // 0x0190
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MusicPlayer_TA");
}

return uClassPointer;
};

void __GFxData_MusicPlayer_TA__OnShellSet_0x3(class UMusicThemeConfig_TA* InConfig);
void __GFxData_MusicPlayer_TA__OnShellSet_0x2(class UMusicUrlConfig_TA* InConfig);
void __GFxData_MusicPlayer_TA__OnShellSet_0x1(class UMusicTrack_TA* _);
void __GFxData_MusicPlayer_TA__StreamerSafeConfigChanged_0x1();
void __GFxData_MusicPlayer_TA__GetTitleTrackBuilder_0x2();
struct FPlaylistTrack __GFxData_MusicPlayer_TA__HandleThemeChange_0x3(struct
FPlaylistTitleTrack X);
class UAkSoundCue* __GFxData_MusicPlayer_TA__HandleThemeChange_0x2(struct
FPlaylistTitleTrack P);
class UAkSoundCue* __GFxData_MusicPlayer_TA__HandleThemeChange_0x1(struct
FPlaylistTitleTrack P);
bool __GFxData_MusicPlayer_TA__HandleSaveDataLoaded_0x2(struct FGfxMusicPlaylist P);
void __GFxData_MusicPlayer_TA__HandleSaveDataLoaded_0x1(struct FPlaylistState P);
void __GFxData_MusicPlayer_TA__SetPlaylistSelected_0x2(class UGfxModal_X* _);
bool __GFxData_MusicPlayer_TA__SetPlaylistSelected_0x1(struct FPlaylistTrack Pt);
static TArray<struct FPlaylistTrack>
__GFxData_MusicPlayer_TA__BuildMasterTracklist_0x2(struct FGfxMusicPlaylist P);
static bool __GFxData_MusicPlayer_TA__BuildMasterTracklist_0x1(struct FGfxMusicPlaylist P);
void __CurrentTrackName__ChangeNotifyFunc();
void UpdateTrackUrl();
void PopLoadingScreenUIState();

```

```

void PushLoadingScreenUIState();
void HandleMusicPlayerStateChanged(class UGameplayMusicPlayer_TA* InGameMusicPlayer,
uint8_t NewState);
void GoToUrl();
void HandlePlaylistsToggled(class UPlayerInput_X* PlayerInput, struct FName ActionName,
unsigned long bEnabled);
static TArray<struct FPlaylistTrack> BuildMasterTracklist(TArray<struct FGFxMusicPlaylist>&
InGFxPlaylists);
void ValidateCurrentTrack();
void SetPlaylistSelected(int32_t Row, unsigned long bSelected);
class UMusicMetrics_TA* GetMetrics();
void HandleTrackEnd(int32_t PlayingID);
void HandleTrackStart(int32_t PlayingID, class FString FileMarker);
void PlayKonamiTheme();
void StopCurrentTrack();
void OnTogglePlaylistSelection();
void TogglePlaylistSelection();
void RebuildActiveTracklist();
void NextTrack();
void PlayCreditsTheme();
void SetPlaylistSelectedByName(struct FName PlaylistName, unsigned long bSelected);
class UIMusicTracklist_TA* BuildTracklist(TArray<struct FPlaylistTrack>& InTracklist);
void HandleSaveDataLoaded(class UMusicPlayerSave_TA* InMusicSave);
void PlayMusicTrack(class UMusicTrack_TA* Track);
class UMusicTrack_TA* PlaylistTrackToMusicTrack(struct FPlaylistTrack Pt);
static TArray<struct FGFxMusicPlaylist> MergePlaylists(TArray<struct FGFxMusicPlaylist>&
OldPlaylists, TArray<struct FGFxMusicPlaylist>& NewPlaylists);
static struct FGFxMusicPlaylist PlaylistToGfxData(class UMusicPlaylist_TA* P);
void HandleThemeChange(class UMusicTheme_TA* InTheme);
class UMusicTracklistFactory_TA* GetTitleTrackBuilder();
void RebuildMasterTracklist();
void StreamerSafeConfigChanged();
void eventOnRemoved();
void eventOnShellSet();
void EventTrackFinished();
};


```

```

// Class TAGame.MusicTracklistFactory_TA
// 0x0030 (0x0060 - 0x0090)
class UMusicTracklistFactory_TA : public UObject
{
public:
class UMusicTracklistNull_TA*           TracklistNull;          // 0x0060 (0x0008)
[0x0000000000000002] (CPF_Const)
class UIMusicTracklistBuilder*          TracklistBuilder_Object; // 0x0068
(0x0008) [0x0000000000000001] (CPF_Edit)
class UIMusicTracklistBuilder*          TracklistBuilder_Interface; // 0x0070
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FScriptDelegate                 LastTrackAction;        // 0x0078 (0x0018)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTracklistFactory_TA");
}

return uClassPointer;
};

class UIMusicTracklist_TA* CreateTracklist(TArray<class UMusicTrack_TA*> InTracklist);
};

// Class TAGame.__GFxData_MusicPlayer_TA__MergePlaylists_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxData_MusicPlayer_TA__MergePlaylists_0x1 : public UObject
{
public:
TArray<struct FGFxMusicPlaylist>           NewPlaylists;           // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_MusicPlayer_TA__MergePlaylists_0x1");
}

return uClassPointer;
};

bool __GFxData_MusicPlayer_TA__MergePlaylists_0x1(struct FGFxMusicPlaylist P);
};

// Class TAGame.MusicPlaylist_TA
// 0x001C (0x0060 - 0x007C)
class UMusicPlaylist_TA : public UObject
{
public:
TArray<struct FPlaylistTrack>           Tracks;           // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UTexture*                         Icon;           // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                           bDefaultSelected : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicPlaylist_TA");
}

return uClassPointer;
};

class FString GetTitle();
};

// Class
TAGame._GFxData_Mutators_TA__InitCustomGameSettingsGetMutatorsByGroupID_0x1
// 0x00104 (0x0060 - 0x007064)
class U_GFxData_Mutators_TA__InitCustomGameSettingsGetMutatorsByGroupID_0x1 : public
UObject
{
public:
TArray<struct FName>int32_t           DisabledCategories;
GroupId;                                // 0x0060 (0x00104) [0x00000000004000000]
(CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Mutators_TA__InitCustomGameSettingsGetMutatorsByGroupID_0x1");
}

return uClassPointer;
};

bool __GFxData_Mutators_TA__InitCustomGameSettings_0x1(class UGameSettingCategory_X*
GSC);
};

// Class TAGame.GFxData_Mutators_TA
// 0x0088 (0x0098 - 0x0120)
class UGFxData_Mutators_TA : public UGFxDataSingleton_X
{
public:
struct FName           SelectedMapName;          // 0x0098 (0x0008)
[0x0000000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t                SelectedGameMode;        // 0x00A0 (0x0004)
[0x0000000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t                SelectedBotDifficulty;
GetMutatorsByGroupID_0x1(struct FCustomGameSetting S);
};

```

```

// Class TAGame.__GFxData_Mutators_TA__InitCustomGameSettings_0x1
// 0x00A410 (0x0004) [0x0000000040000001] (CPF_Edit | CPF_EditInlineNotify)
TArray<struct FGameModeData>           GameModes;                      // 0x00A8
(0x0010) [0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FCustomGameSetting>        CustomGameSettings;          // 0x00B8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<class UGameSettingCategory_X*>    GameSettingCategories;      // 0x00C8 (0x0010) [0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UGameSetting_X*>             GameModeMutatorSettingPresets; // 0x00D8 (0x0010) [0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                                PresetCategoryIndex;          // 0x00E8 (0x0004)
[0x0000004000000000]

TArray<struct FModeMapPair>            ModeMaps; 60 - 0x0070
class U_GFxDATA_Mutators_TA__InitCustomGameSettings_0x1 : public UObject
{
public:
TArray<struct FName>                  DisabledCategories;          // 0x00F60
(0x0010) [0x00000000004020000] (CPF_Transient | CPF_NeedCtorLink)
uint8_t                               SettingsType;                // 0x0100 (0x0001)
[0x0000000000002000] (CPF_Transient)
class UProfile_TA*                   Profile;                  // 0x0108 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGameMode_TA*                 GameModeObj;              // 0x0110 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGameTags_TA*                 GameTags;                 // 0x0118 (0x0008)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Mutators_TA");
}

return uClassPointer;
};

void __GFxDATA_Mutators_TA__SetCustomGameSetting_0x2(struct FCategorySettingPair P);
bool __GFxDATA_Mutators_TA__SeNeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Mutators_TA__InitCustomGameSettings_0x1(struct FCategorySettingPair
P);
void SetCustomGameSetting(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex);

```

```
bool ResetInvalidSettings(int32_t MutatorCategoryIndex);
bool IsValidSetting(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex, unsigned long bCoerceMap);
class FString GetCustomGameSettingDescription(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex);
void ForceClearAllTags();
void SetSelectedBotDifficulty(int32_t BotDifficulty);
class FString GetGameTagValueFromIndex(class FString GameTagCategory, int32_t Index);
int32_t GetGameTagIndex(class FString GameTagCategory, class FString Value);
void SetSelectedGameMode(int32_t GameMode);
void SetSelectedMapName(struct FName MapName);
void InitCustomGameSettings();
void RefreshCustomGameSettings(int32_t MutatorCategoryIndex, int32_t MutatorSettingIndex);
void OnUpdatedSettings();
void PopulateGameModes();
void SetCustomMatchSettings(struct FCustomMatchSettings& InSettings);
struct FCustomMatchSettings GetCustomMatchSettings();
void SetGameTag(class FString Key, class FString Value);
class FString GetGameTag(class FString Key);
void InitMutatorsFromTournament(uint64_t TournamentID);
void InitMutators(uint8_t InSettingsType);
void eventOnShellSet();
class UOnlineGame_TA* GetOnlineGame(");
}

return uClassPointer;
};

bool __GFxData_Mutators_TA__InitCustomGameSettings_0x1(class UGameSettingCategory_X* GSC);
};

// Class TAGame.__GFxData_NotificationManager_TA__HandleNotificationHidden_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_NotificationManager_TA__HandleNotificationHidden_0x1 : public UObject
{
public:
class UNotification_TA* Notification; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_NotificationManager_TA__HandleNotificationHidden_0x1");
}

return uClassPointer;
};
```

```
bool __GFxData_NotificationManager_TA_HandleNotificationHidden_0x1(class UNotification_TA* N);
};

// Class TAGame.GFxData_NotificationManager_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_NotificationManager_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFXNotificationInfo> Notifications; // 0x0098 (0x0010)
[0x0000004040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bShowInGameNotifications : 1; // 0x00A8 (0x0004)
[0x0000004040000001] [0x00000001] (CPF_Edit | CPF_EditInlineNotify)
unsigned long bShowItemShopNotifications : 1; // 0x00A8 (0x0004)
[0x0000004040000001] [0x00000002] (CPF_Edit | CPF_EditInlineNotify)
class UNotificationSave_TA* NotificationSave; // 0x00B0 (0x0008)
[0x0000000000002000] (CPF_Transient)
TArray<class UNotification_TA*> PendingNotifications; // 0x00B8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UNotification_TA*> ActiveNotifications; // 0x00C8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_NotificationManager_TA");
}

return uClassPointer;
};

void __GFxData_NotificationManager_TA_RefreshNotifications_0x1(class UNotification_TA* Notification);
void ResumeNotificationTimer(int32_t NotificationID);
void PauseNotificationTimer(int32_t NotificationID, unsigned long bShouldHide);
void OnNotificationHidden(int32_t NotificationID);
void DismissNotification(int32_t NotificationID);
void RejectNotification(int32_t NotificationID);
void AcceptNotification(int32_t NotificationID);
int32_t GetNotificationRowIndex(int32_t NotificationID);
void HandleModalClosed();
void HandleModalOpened();
int32_t GetNotificationInfoIndex(int32_t NotificationID);
void eventOnRemoved();
void DestroyGFxNotifications();
class UNotificationManager_TA* GetNotificationManager();
void HandleNotificationRemoved(class UNotificationManager_TA* NotificationManager, class UNotification_TA* Notification);
void HandleNotificationHidden(class UNotification_TA* Notification);
void HandleNotificationShown(class UNotification_TA* Notification);
```

```
void ProcessPendingNotifications();
void HandleNotificationAdded(class UNotificationManager_TA* NotificationManager, class
UNotification_TA* Notification);
void RefreshNotifications();
void HandleNotificationSaveUpdated();
void HandleNotificationSaveAdded(class UNotificationSave_TA* InNotificationSave);
void HandleAccountSettingsAdded(class UAccountSettingsComponent_TA* AccountSettings);
void eventOnShellSet();
};

// Class TAGame.__GFxData_NotificationManager_TA__HandleNotificationRemoved_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_NotificationManager_TA__HandleNotificationRemoved_0x1 : public UObject
{
public:
    class UNotification_TA*           Notification;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_NotificationManager_TA__HandleNotificationRemoved_0x1");
        }

        return uClassPointer;
    }

    bool __GFxData_NotificationManager_TA__HandleNotificationRemoved_0x2(class
UNotification_TA* N);
    bool __GFxData_NotificationManager_TA__HandleNotificationRemoved_0x1(struct
FGFXNotificationInfo N);
};

// Class TAGame.__GFxData_NotificationManager_TA__OnNotificationHidden_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GFxData_NotificationManager_TA__OnNotificationHidden_0x1 : public UObject
{
public:
    int32_t                          NotificationID;        // 0x0060 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
```

```

TAGame.__GFxData_NotificationManager_TA__OnNotificationHidden_0x1");
}

return uClassPointer;
};

bool __GFxData_NotificationManager_TA__OnNotificationHidden_0x1(class UNotification_TA*
N);
};

// Class TAGame.__GFxData_Party_TA__ActivatePartyUp_0x3
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Party_TA__ActivatePartyUp_0x3 : public UObject
{
public:
class AStayAsPartyVoter_TA*           PartyVoter;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA__ActivatePartyUp_0x3");
}

return uClassPointer;
};

void __GFxData_Party_TA__ActivatePartyUp_0x3(class UGFxModal_X* _);

// Class TAGame.StayAsPartyVoter_TA
// 0x0070 (0x0268 - 0x02D8)
class AStayAsPartyVoter_TA : public AActor
{
public:
struct FUniqueNetId           PlayerID;          // 0x0268 (0x0048)
[0x0000000000400020] (CPF_Net | CPF_NeedCtorLink)
TArray<struct FUniqueNetId>   Followers;        // 0x02B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class AStayAsPartyVoteYes_TA* YesVote;          // 0x02C0 (0x0008)
[0x0000000000000000]
unsigned long                 bSentPartyInvites : 1; // 0x02C8 (0x0004)
[0x0000000000000000] [0x00000001]
class APRI_TA*                PRI;              // 0x02D0 (0x0008)
[0x0000000100002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StayAsPartyVoter_TA");
}

return uClassPointer;
};

bool __StayAsPartyVoter_TA__GetPartyUpError_0x4(class APRI_TA* TeammatePRI);
bool __StayAsPartyVoter_TA__GetPartyUpError_0x3(class APRI_TA* P);
bool __StayAsPartyVoter_TA__GetPartyUpError_0x2(class APRI_TA* P);
bool __StayAsPartyVoter_TA__GetPartyUpError_0x1(class APRI_TA* P);
void ServerJoinError(class UError* ReportedError);
void ReportJoinError(class UError* ReportedError);
void ServerStayAsPartyVote();
bool CanPlayerPartyUpServer();
void VoteYes();
bool GetPartyUpError(class FString& Error);
void RemovePartyFollower(struct FUniqueNetId& OldFollower);
void AddPartyFollower(struct FUniqueNetId& NewFollower);
void SetPartyFollowers(TArray<struct FUniqueNetId>& PartyFollowers);
void Setup(struct FUniqueNetId InPlayerId, TArray<struct FUniqueNetId>& PartyFollowers);
void eventDestroyed();
void RefreshORSGroup();
void eventOnOwnerChanged();
static class AStayAsPartyVoter_TA* SpawnVoter(class AActor* VoterOwner, struct
FUniqueNetId& VoterId, TArray<struct FUniqueNetId>& PartyFollowers);
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.GFxData_Party_TA
// 0x0070 (0x0098 - 0x0108)
class UGFxData_Party_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGfxPartyMember> Members; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t MaxPartySize; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
struct FName Processing; // 0x00AC (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
class UOnlineGameParty_X* OnlineParty; // 0x00B8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UPersonas_TA* PersonasData; // 0x00C0 (0x0008)
[0x0000804000002000] (CPF_Transient)
class UOnlineClubCache_X* Clubs; // 0x00C8 (0x0008)
[0x0001800004080008] (CPF_ExportObject | CPF_Component | CPF>EditInline)
class UVanitySetManager_TA* VanityManager; // 0x00D0
(0x0008) [0x0000800000000000]
uint8_t PsyNetPartyStatus; // 0x00D8 (0x0001)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString RegionRestrictedPlayerTrade; // 0x00E0 (0x0010)

```

```

[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UGFxModal_X*                               PartyErrorModal;           // 0x00F0 (0x0008)
[0x0000000000000000]
unsigned long                                     bPendingAllowTrade : 1;    // 0x00F8 (0x0004)
[0x0001000000000000] [0x00000001]
class UCrossplayConfig_X*                        CrossplayConfig;         // 0x0100 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Party_TA");
}

return uClassPointer;
};

void __GFxData_Party_TA__OnShellSet_0x1(class USaveGameManager_TA* M, class
USaveData_TA* S, class UError* E);
struct FUniqueNetId __GFxData_Party_TA__ActivatePartyUp_0x2(struct FGFxPartyMember M);
struct FUniqueNetId __GFxData_Party_TA__ActivatePartyUp_0x1(class APRI_TA* T);
struct FUniqueNetId __GFxData_Party_TA__RebuildPartyData_0x1(struct FPartyMember
PMember);
void OnPartyMemberLeftMatch(struct FUniqueNetId PartyMemberID);
void HandleFollowTheLeaderToMainMenu(class UGFxModal_X* Modal);
void HandleLeaderChange(class UOnlineGameParty_X* PartyObject, struct FUniqueNetId
NewLeader);
void HandleLeaderLeftOnlineGame(class UOnlineGameParty_X* PartyObject);
void ClearPartyJoinGameDelegates();
void HandlePartyJoinGameSuccess(class UOnlineGameParty_X* PartyObject);
void HandlePartyJoinGameError(class UOnlineGameParty_X* PartyObject, class FString
LocalizedString);
void HandlePartyJoinGameConfirmed(class UGFxModal_X* Modal);
void HandlePartyJoinGameConfirmation(class UOnlineGameParty_X* PartyObject);
void HandlePartyError(class UOnlineGameParty_X* PartyObject, class UError* Error);
void HandleReceivedPartyPersonalInfo(class URPC_GetPersonalInfo_TA* RPC);
int32_t SortMembersCallback(struct FGFxPartyMember A, struct FGFxPartyMember B);
void RebuildPartyData();
void RebuildPartyDataDelayed();
void OnPartyChanged(class UOnlineGameParty_X* PartyObject);
bool ProcessInvitationResponse(class UOnlineGameParty_X* Party, struct FScriptDelegate
JoinPartyCallback, struct FUniqueLobbyId& PartyID, struct FUniqueNetId& InviterId, class
UGFxModal_X*& CreatedModal);
void OnPlayerInvitedSilent(class UOnlineGameParty_X* Party, struct FScriptDelegate
JoinPartyCallback, struct FUniqueLobbyId& PartyID, struct FUniqueNetId& InviterId);
void JoinAttempt(class UOnlineGameParty_X* Party, struct FScriptDelegate JoinPartyCallback,
struct FUniqueLobbyId& PartyID, struct FUniqueNetId& InviterId);
void OnPlayerInvited(class UOnlineGameParty_X* Party, struct FScriptDelegate
JoinPartyCallback, struct FUniqueLobbyId& PartyID, struct FUniqueNetId& InviterId);
void HandleGetAvatar(struct FUniqueNetId InPlayerId, class UTexture* Avatar, class FString

```

```

OnlinePlayerName);
void SetLeader(int32_t Index);
void OnPartyMembersPersonaUpdated();
void SetSearchingStatus(unsigned long bSearching, uint8_t StatusOwner);
bool IsInSameMatch(int32_t PartyMemberIdx);
void HandleJoinFriend();
void JoinFriend(int32_t PartyMemberIdx);
void JoinPartyMatch();
bool IsPartyMatchAvailable();
bool IsPlayerJoinable(int32_t PartyMemberIdx);
void HandlePlayerInMatchChanged(class UOnlineGameParty_X* PartyObject, struct FUniqueNetId InMemberId);
void KickPlayer(int32_t Index);
void ShowInviteUI();
bool CanSendPartyInviteToUser(struct FUniqueNetId PlayerID);
void OnConfirmLeaveParty(class UGFxModal_X* Modal);
void LeaveParty();
void OnPartyCreatedWithoutInvite();
void OnConfirmCreatePlatformParty(class UGFxModal_X* Modal);
void AskCreatePlatformParty();
void OnPartyCreated(class UOnlineGameParty_X* Party, unsigned long bWasSuccessful);
void CreateParty();
bool IsProcessing();
void HandleProcessingStateChanged(class UOnlineGameParty_X* P);
void HandleProfileSet(class ULocalPlayer_TA* LocalPlayer);
void eventOnRemoved();
uint8_t GetPsyNetPartyStatus();
bool ActivatePartyUp();
void UpdatePsyNetPartyStatus();
void HandlePerConChanged(class UPerCon_X* PerCon);
void HandleConnectionChanged(class UPsyNetConnection_X* Connection);
void HandlePrimaryPlayerChanged(class UOnlineGameAccount_X* InAccount, struct FUniqueNetId PlayerID);
void HandlePsyNetConfigChanged(class UPsyNetConfig_X* Config);
void HandleClubUpdated(class UOnlineClubCache_X* Cache, class UClubDetails_X* ClubDetails);
void UpdateSaveObjectReferences();
void eventOnShellSet();
};

// Class TAGame.__GFxDATA_Party_TA__HandleAcceptInviteToTrade_0x1
// 0x0050 (0x0060 - 0x00B0)
class U__GFxDATA_Party_TA__HandleAcceptInviteToTrade_0x1 : public UObject
{
public:
    class UGFxModal_X*                               Modal;                      // 0x0060 (0x0008)
    [0x0000000000000000]                           Inviter;                   // 0x0068 (0x0048)
    struct FUniqueNetId [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA__HandleAcceptInviteToTrade_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__GFxData_Party_TA__HandleInviteToTrade_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxDATA_Party_TA_HandleInviteToTrade_0x1 : public UObject
{
public:
struct FUniqueNetId           Inviter;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA__HandleInviteToTrade_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__GFxData_Party_TA__HandlePartyError_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_Party_TA_HandlePartyError_0x1 : public UObject
{
public:
class UError*                 Error;             // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA__HandlePartyError_0x1");
}
```

```

return uClassPointer;
};

void __GFxData_Party_TA_HandlePartyError_0x1(class UGFxModal_X* Modal);
};

// Class TAGame.__GFxData_Party_TA_InvitePlayerToTrade_0x1
// 0x0004 (0x0060 - 0x0064)
class U_GFxData_Party_TA_InvitePlayerToTrade_0x1 : public UObject
{
public:
int32_t Index; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA_InvitePlayerToTrade_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__GFxData_Party_TA_OnPlayerInvited_0x1
// 0x0078 (0x0060 - 0x00D8)
class U_GFxData_Party_TA_OnPlayerInvited_0x1 : public UObject
{
public:
class UOnlineGameParty_X* Party; // 0x0060 (0x0008)
[0x0000000000000000]
struct FUniqueLobbyId PartyID; // 0x0068 (0x0010)
[0x0000000000000000]
struct FUniqueNetId InviterId; // 0x0078 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate JoinPartyCallback; // 0x00C0 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA_OnPlayerInvited_0x1");
}
}

```

```

return uClassPointer;
};

void __GFxData_Party_TA__OnPlayerInvited_0x1(class UGFxModal_X* _);
};

// Class TAGame.__GFxData_Party_TA__ProcessInvitationResponse_0x1
// 0x0028 (0x0060 - 0x0088)
class U_GFxData_Party_TA__ProcessInvitationResponse_0x1 : public UObject
{
public:
    struct FScriptDelegate           JoinPartyCallback;          // 0x0060 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FUniqueLobbyId           PartyID;                  // 0x0078 (0x0010)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Party_TA__ProcessInvitationResponse_0x1");
        }
    }

    return uClassPointer;
};

void __GFxData_Party_TA__ProcessInvitationResponse_0x1(class UGFxModal_X* _);
};

// Class TAGame.__GFxData_Party_TA__SendAllowTradeRPC_0x1
// 0x0078 (0x0060 - 0x00D8)
class U_GFxData_Party_TA__SendAllowTradeRPC_0x1 : public UObject
{
public:
    struct FScriptDelegate           OnSuccess;                // 0x0060 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FUniqueNetId              OtherPlayerNetId;       // 0x0078 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           OnFail;                  // 0x00C0 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class

```

```

TAGame.__GFxData_Party_TA__SendAllowTradeRPC_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1
// 0x0018 (0x0060 - 0x0078)
class U_GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1 : public UObject
{
public:
struct FUniqueLobbyId           InLobby;          // 0x0060 (0x0010)
[0x0001000000000000]
class UPersona_TA*              Persona;         // 0x0070 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1");
}

return uClassPointer;
};

void
__GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1__GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1_0x2(class UNotification_TA* Notification);
void
__GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1__GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1_0x1(class UNotification_TA* Notification);
void __GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x1(class UPersona_TA*
InPersona);
};

// Class TAGame.PartyInviteNotification_TA
// 0x0058 (0x0170 - 0x01C8)
class UPartyInviteNotification_TA : public UNotification_TA
{
public:
struct FUniqueLobbyId           LobbyID;          // 0x0170 (0x0010)
[0x0000000000000000]
struct FUniqueNetId              PlayerID;         // 0x0180 (0x0048)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyInviteNotification_TA");
}

return uClassPointer;
};

bool ShouldShow();
struct FUniqueNetId GetSenderID();
class UPartyInviteNotification_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
class UPartyInviteNotification_TA* SetLobbyID(struct FUniqueLobbyId InLobbyId);
};

// Class TAGame.GFxData_PartyInvite_TA
// 0x0068 (0x0098 - 0x0100)
class UGFxData_PartyInvite_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FInviteDatabind> Invites; // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FInviteTarget> PendingInvites; // 0x00A8 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FUniqueNetId> SilentSentInvites; // 0x00B8 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UParties_X* PsyNetParty; // 0x00C8 (0x0008)
[0x0001800000000000]
class UOnlineGameParty_X* OnlineGameParty; // 0x00D0
(0x0008) [0x0001800000000000]
class UCrossplayConfig_X* CrossplayConfig; // 0x00D8 (0x0008)
[0x0001800000000001] (CPF>Edit)
class UPersonas_TA* PersonasData; // 0x00E0 (0x0008)
[0x0001804000002000] (CPF_Transient)
struct FScriptDelegate __SendPartyInvitation__Delegate; // 0x00E8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PartyInvite_TA");
}

return uClassPointer;
};

class FString __GFxData_PartyInvite_TA__InviteToPartyInternal_0x3(struct FPartyMember M);
bool __GFxData_PartyInvite_TA__InviteToPartyInternal_0x2(struct FPartyMember Member);

```

```

void __GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x3(class UNotification_TA*
Notification);
void __GFxData_PartyInvite_TA__HandlePlayerInvitePending_0x2(class UNotification_TA*
Notification);
bool __GFxData_PartyInvite_TA__HandlePartyChanged_0x2(class UPartyInviteNotification_TA*
N);
bool __GFxData_PartyInvite_TA__HandlePartyChanged_0x1(struct FInviteDatabind I);
void HandleBlockStatusChanged(class UPersona_TA* Persona);
void ShowPlatformInviteUI();
bool CanShowPlatformInviteUI();
void HandlePartyChanged(class UOnlineGameParty_X* Party);
void HandlePlayerInvitePending(struct FUniqueLobbyId InLobby, struct FUniqueNetId
InFromUser, class FString InFromName);
void OnInvited();
void SetUserInvitedResponse(int32_t InviteIndex, unsigned long bAccepted);
void HandlePartyInviteNotificationResponse(struct FUniqueNetId InPlayerId, unsigned long
bAccepted);
void OnPartyCreatedFromInvite();
void OnJoinLobbyComplete(unsigned long bWasSuccessful, class FString Error, struct
FActiveLobbyInfo& LobbyInfo, struct FUniqueLobbyId& LobbyId);
void OnPartyCreated(class UOnlineGameParty_X* OnlineParty, unsigned long bWasSuccessful);
void ShowPartyInviteError(class FString ErrorMessage);
void OnPartyInviteComplete(unsigned long bSucceeded, struct FUniqueNetId InviteID, class
UError* InError);
void InviteToPartyInternal(struct FUniqueNetId ToPlayer, struct FScriptDelegate InviteHandler);
void InviteToParty(struct FUniqueNetId PlayerID);
void InviteToPartySilent(struct FUniqueNetId PlayerID);
void ShowInviteUI();
void eventOnRemoved();
void eventOnShellSet();
void SendPartyInvitation(struct FUniqueNetId ToPlayer);
};

// Class TAGame.__GFxData_PartyInvite_TA__InviteToPartyInternal_0x1
// 0x0060 (0x0060 - 0x00C0)
class U__GFxData_PartyInvite_TA__InviteToPartyInternal_0x1 : public UObject
{
public:
    struct FUniqueNetId           ToPlayer;          // 0x0060 (0x0048)
    [0x0001000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate        InviteHandler;     // 0x00A8 (0x0018)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_PartyInvite_TA__InviteToPartyInternal_0x1");
        }
    }
}

```

```

return uClassPointer;
};

void __GFxData_PartyInvite_TA__InviteToPartyInternal_0x4(class UPrivilegeCheck_X*
PrivilegeCheck);
bool __GFxData_PartyInvite_TA__InviteToPartyInternal_0x1(struct FPartyMember Member);
};

// Class TAGame.__GFxData_PartyInvite_TA__InviteToPartySilent_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_PartyInvite_TA__InviteToPartySilent_0x1 : public UObject
{
public:
class UParties_X*           PsyNetInterface;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_PartyInvite_TA__InviteToPartySilent_0x1");
}

return uClassPointer;
};

void __GFxData_PartyInvite_TA__InviteToPartySilent_0x1(struct FUniqueNetId PlayerID);
};

// Class
TAGame.__GFxData_PlayerAvatarBorderPreview_TA__CreatePlayerAvatarBorderPreview_0x1
// 0x0014 (0x0060 - 0x0074)
class U_GFxData_PlayerAvatarBorderPreview_TA__CreatePlayerAvatarBorderPreview_0x1 :
public UObject
{
public:
class UPlayerAvatarBorder_TA*      AvatarBorderPreview;    // 0x0060
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UOnlineProduct_TA*          OnlineProduct;        // 0x0068 (0x0008)
[0x0000000000000000]
int32_t                           InColorID;           // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class
TAGame._GFxData_PlayerAvatarBorderPreview_TA__CreatePlayerAvatarBorderPreview_0x1");
}

return uClassPointer;
};

void __GFxData_PlayerAvatarBorderPreview_TA__CreatePlayerAvatarBorderPreview_0x1(class
UPlayerVanity_TA* PlayerVanity);
};

// Class TAGame.GFxData_PlayerAvatarBorderPreview_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_PlayerAvatarBorderPreview_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFxPlayerAvatarBorderRef>      AvatarBorderPreviews;           // 0x0098
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerAvatarBorderPreview_TA");
}

return uClassPointer;
};

int32_t GetBorderIndexByHashID(struct FProductHashID HashID);
int32_t GetPlayerAvatarBorder(struct FUniqueNetId PlayerID);
void DestroyPlayerAvatarBorderPreview(struct FProductHashID HashID);
struct FProductInstanceId CreatePlayerAvatarBorderPreview(struct FProductHashID HashID,
int32_t InColorID);
};

// Class TAGame._GFxData_PlayerAvatarPreview_TA__CreatePlayerAvatarPreview_0x1
// 0x0014 (0x0060 - 0x0074)
class U_GFxData_PlayerAvatarPreview_TA__CreatePlayerAvatarPreview_0x1 : public UObject
{
public:
class UPlayerAvatar_TA*          AvatarPreview;           // 0x0060 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UOnlineProduct_TA*         OnlineProduct;        // 0x0068 (0x0008)
[0x0000000000000000]
int32_t                          InColorID;            // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_PlayerAvatarPreview_TA__CreatePlayerAvatarPreview_0x1");
}

return uClassPointer;
};

void __GFxData_PlayerAvatarPreview_TA__CreatePlayerAvatarPreview_0x1(class
UPlayerVanity_TA* PlayerVanity);
};

// Class TAGame.GFxData_PlayerAvatarPreview_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_PlayerAvatarPreview_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFxPlayerAvatarRef>           AvatarPreviews;          // 0x0098
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerAvatarPreview_TA");
}

return uClassPointer;
};

int32_t GetAvatarIndexByHashID(struct FProductHashID HashID);
int32_t GetPlayerAvatar(struct FUniqueNetId PlayerID);
void DestroyPlayerAvatarPreview(struct FProductHashID HashID);
struct FProductInstanceId CreatePlayerAvatarPreview(struct FProductHashID HashID, int32_t
InColorID);
};

// Class TAGame.ProductHash_TA
// 0x0000 (0x0060 - 0x0060)
class UProductHash_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductHash_TA");
}

return uClassPointer;
};

static struct FProductHashID GetHashIDFromInstanceID(struct FProductInstanceID InstanceID,
class UOnlineProductStoreSet_TA* ProductSet);
static class UProduct_TA* GetProductFromInstanceID(struct FProductInstanceID InstanceID,
class UOnlineProductStoreSet_TA* ProductSet);
static struct FProductHashSource GetProductSourceFromInstanceID(struct FProductInstanceID InstanceID,
class UOnlineProductStoreSet_TA* ProductSet);
static TArray<class UOnlineProduct_TA*> GetOnlineProductsFromHashID(struct FProductHashID HashID,
class UOnlineProductStoreSet_TA* ProductSet);
static class UOnlineProduct_TA* GetOnlineProductFromHashID(struct FProductHashID HashID,
class UOnlineProductStoreSet_TA* ProductSet);
static class UProduct_TA* GetProductFromHashID(struct FProductHashID HashID, class
UOnlineProductStoreSet_TA* ProductSet);
static int32_t GetProductIDFromHashID(struct FProductHashID HashID, class
UOnlineProductStoreSet_TA* ProductSet);
static struct FProductHashSource ProductToProductSource(class UProduct_TA* Product);
static struct FProductHashSource OnlineProductToProductSource(class UOnlineProduct_TA*
OnlineProduct);
static TArray<struct FProductHashSource> GetProductSourcesFromHashID(struct
FProductHashID HashID, class UOnlineProductStoreSet_TA* ProductSet);
static struct FProductHashSource GetProductSourceFromHashID(struct FProductHashID HashID,
class UOnlineProductStoreSet_TA* ProductSet);
};

// Class TAGame.__GFxData_PlayerBanners_TA__CreatePlayerBanner_0x1
// 0x002C (0x0060 - 0x008C)
class U__GFxData_PlayerBanners_TA__CreatePlayerBanner_0x1 : public UObject
{
public:
class UPlayerBanner_TA* Banner; // 0x0060 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FProductHashSource ProductSource; // 0x0068 (0x0020)
[0x0000000000000000]
int32_t InColorID; // 0x0088 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_PlayerBanners_TA__CreatePlayerBanner_0x1");
}
}

```

```

return uClassPointer;
};

void __GFxData_PlayerBanners_TA__CreatePlayerBanner_0x1(class UPlayerVanity_TA*
PlayerVanity);
};

// Class TAGame.GFxData_PlayerBanners_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_PlayerBanners_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGfxPlayerBannerRef>           Banners;                      // 0x0098 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UPlayerBanner_TA*             LastPreviewBanner;                // 0x00A8 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerBanners_TA");
}

return uClassPointer;
};

int32_t GetBannerIndexByHashID(struct FProductHashID HashID);
int32_t GetPlayerBanner(struct FUniqueNetId PlayerID);
void DestroyPendingAndActivePlayerBannerPreviews();
void DestroyPlayerBannerPreview();
void DestroyPlayerBanner(struct FProductHashID HashID);
void DestroyBannerByNetID(struct FUniqueNetId PlayerID);
void CreatePlayerBannerPreview(struct FProductHashID HashID, int32_t InColorID);
void OnBannerPreviewLoaded(struct FProductInstanceId InstanceID);
class UProfile_TA* GetProfile();
void HandleBannerPreviewLoaded(class UPlayerBanner_TA* Banner, class UOnlineProduct_TA*
OnlineProduct, int32_t InColorID, struct FUniqueNetId PlayerID, struct FProductHashID HashID);
int32_t CreatePlayerBanner(struct FProductHashID HashID, int32_t InColorID);
};

// Class TAGame.__GFxData_PlayerBanners_TA__CreatePlayerBannerPreview_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxData_PlayerBanners_TA__CreatePlayerBannerPreview_0x1 : public UObject
{
public:
class UOnlineProduct_TA*           OnlineProduct;                  // 0x0060 (0x0008)
[0x0000000000000000]
int32_t                           InColorID;                    // 0x0068 (0x0004)
[0x0000000000000000]
struct FProductHashID            HashID;                      // 0x006C (0x0004)

```

```
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_PlayerBanners_TA__CreatePlayerBannerPreview_0x1");
}

return uClassPointer;
};

void __GFxData_PlayerBanners_TA__CreatePlayerBannerPreview_0x1(class UPlayerVanity_TA*
PlayerVanity);
};

// Class TAGame._GFxData_Playlist_TA_SetPlayerCount_0x1
// 0x0004 (0x0060 - 0x0064)
class U_GFxData_Playlist_TA_SetPlayerCount_0x1 : public UObject
{
public:
int32_t           Count;           // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Playlist_TA_SetPlayerCount_0x1");
}

return uClassPointer;
};

bool __GFxData_Playlist_TA_SetPlayerCount_0x1(int32_t X);
};

// Class TAGame.GFxData_Playlist_TA
// 0x00C4 (0x0094 - 0x0158)
class UGFxData_Playlist_TA : public UGFxDataRow_X
{
public:
class UGameSettingPlaylist_X*      Playlist;          // 0x0098 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t           Id;               // 0x00A0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
```

```

class FString           KeyName;          // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           FriendlyName;      // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           Description;       // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           BadgeTitle;        // 0x00D8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                TeamSize;         // 0x00E8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
unsigned long          bStandardMatch : 1;    // 0x00EC (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long          bRanked : 1;        // 0x00EC (0x0004)
[0x0000000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long          bSolo : 1;         // 0x00EC (0x0004)
[0x0000000040000000] [0x00000004] (CPF>EditInlineNotify)
unsigned long          bSelected : 1;     // 0x00EC (0x0004)
[0x0000000040000000] [0x00000008] (CPF>EditInlineNotify)
unsigned long          bEnabled : 1;       // 0x00EC (0x0004)
[0x0000000040000000] [0x00000010] (CPF>EditInlineNotify)
unsigned long          bRestricted : 1;   // 0x00EC (0x0004)
[0x0000000040000000] [0x00000020] (CPF>EditInlineNotify)
unsigned long          bNew : 1;          // 0x00EC (0x0004)
[0x0000000040000000] [0x00000040] (CPF>EditInlineNotify)
unsigned long          bIsMicroEventPlaylist : 1; // 0x00EC (0x0004)
[0x0000000040000000] [0x00000080] (CPF>EditInlineNotify)
unsigned long          bTournament : 1;     // 0x00EC (0x0004)
[0x0000000040000000] [0x00000100] (CPF>EditInlineNotify)
unsigned long          bCompletedPlacementMatches : 1; // 0x00EC
(0x0004) [0x0000000040000000] [0x00000200] (CPF>EditInlineNotify)
unsigned long          bOpenDetailsOnFirstTimeClicked : 1; // 0x00EC
(0x0004) [0x0000000040000000] [0x00000400] (CPF>EditInlineNotify)
unsigned long          bDisableSaveReplays : 1;      // 0x00EC (0x0004)
[0x0000000040000000] [0x00000800] (CPF>EditInlineNotify)
unsigned long          bPlaylistAllowsStayAsParty : 1; // 0x00EC (0x0004)
[0x0000000040000000] [0x00001000] (CPF>EditInlineNotify)
unsigned long          bUndersizedPartyWithRankDisparity : 1; // 0x00EC
(0x0004) [0x0001000040000000] [0x00002000] (CPF>EditInlineNotify)
unsigned long          bIsTimeRestricted : 1;        // 0x00EC (0x0004)
[0x0000000040000000] [0x00004000] (CPF>EditInlineNotify)
unsigned long          bForceDetailsPage : 1;       // 0x00EC (0x0004)
[0x0000000040000000] [0x00008000] (CPF>EditInlineNotify)
class UTexture*        PlaylistBackgroundTexture; // 0x00F0 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
class UTexture*        PlaylistIconActiveTexture; // 0x00F8 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
class UTexture*        PlaylistIconInactiveTexture; // 0x0100 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
class UTexture*        PlaylistNodeThumbnailTexture; // 0x0108 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
uint8_t                PlayerCountBucket;      // 0x0110 (0x0001)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t                SkillTier;           // 0x0114 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)

```

```
int32_t SkillDivision; // 0x0118 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t SkillMatchesPlayed; // 0x011C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t SkillPlacementMatchesPlayed; // 0x0120 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float SkillIMMR; // 0x0124 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t PrevSkillTier; // 0x0128 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t PrevSkillDivision; // 0x012C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t MaximumRankDisparity; // 0x0130 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t StartTime; // 0x0138 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
uint64_t EndTime; // 0x0140 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t DetailsGroup; // 0x0148 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UMicroEventConfig_TA* MicroEventConfig; // 0x0150
(0x0008) [0x0000000000002000] (CPF_Transient)
```

public:

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Playlist_TA");
}
```

```
return uClassPointer;
};
```

```
void __GFxData_Playlist_TA_SetPlaylist_0x3(class UOnlineGameParty_X* Party, struct
FUniqueNetId MemberId);
void __GFxData_Playlist_TA_SetPlaylist_0x2(class UOnlineGameParty_X* Party, struct
FUniqueNetId NewLeader);
void __GFxData_Playlist_TA_SetPlaylist_0x1(class UOnlineGameParty_X* Party, struct
FUniqueNetId MemberId);
void __GFxData_Playlist_TA_HandleMenuTreePlaylistManagerChanged_0x1(class
UMenuTreePlaylistManager_TA* InManager);
void __GFxData_Playlist_TA_DownloadPlaylistImages_0x2(class UTexture2DDynamic*
ThumbnailTexture);
void __GFxData_Playlist_TA_DownloadPlaylistImages_0x1(class UTexture2DDynamic*
BackgroundTexture);
void MarkDetailsPageSeen();
void UpdatePlaylistTimeWindow();
void UpdateUndersizedParty();
void UpdateSkillData();
void HandleSkillUpdateOnGameJoin(unsigned long bSuccess, class FString FailReason);
void SetPlayerCount(int32_t Count);
```

```

void SetMicroEventImage();
void HandleMicroEventChanged(class UMicroEventConfig_TA* InMicroEventConfig);
void DownloadPlaylistImages();
void HandleMenuTreePlaylistManagerChanged(class UMenuTreePlaylistManager_TA* InPlaylistManager);
void HandleProfileSet(class ULocalPlayer_TA* LocalPlayer);
void SetDefaultPlaylistImage();
void SetPlaylist(class UGameSettingPlaylist_X* InPlaylist);
class UOnlineGameParty_X* GetOnlineGameParty();
class UOnlineGameSkill_X* GetOnlineGameSkill();
};

// Class TAGame.__GFxData_PRI_TA__HandleOwnerChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_PRI_TA__HandleOwnerChanged_0x1 : public UObject
{
public:
class UProfileLoadoutSave_TA*           LoadoutSave;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_PRI_TA__HandleOwnerChanged_0x1");
}

return uClassPointer;
};

void __GFxData_PRI_TA__HandleOwnerChanged_0x1();
};

// Class TAGame.ProfileLoadoutSave_TA
// 0x00C8 (0x00C8 - 0x0190)
class UProfileLoadoutSave_TA : public USaveObject_TA
{
public:
TArray<class ULoadoutSet_TA*>           LoadoutSets;          // 0x00C8
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)
class ULoadoutSet_TA*                   EquippedLoadoutSet;    // 0x00D8 (0x0008)
[0x0000000000000000]
struct FName                           PlayerTitle;         // 0x00E0 (0x0008)
[0x0008000000000000]
int32_t                                PreviewTeamIndex;     // 0x00E8 (0x0004)
[0x0000000000000000]
struct FProductInstanceId             MusicStingerInstanceId; // 0x00F0
(0x0010) [0x0008000000000000]
struct FScriptDelegate                _EventEquippedLoadoutChanged__Delegate; // 0x0100 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

```

```

struct FScriptDelegate           __EventProductEquipped__Delegate;      // 0x0118
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventOnlineProductEquipped__Delegate; // 0x0130
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPreviewTeamChanged__Delegate;   // 0x0148
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate           __PlayerTitle__ChangeNotify;        // 0x0160
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate           __MusicStingerInstanceID__ChangeNotify; // 0x0178
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfileLoadoutSave_TA");
}

return uClassPointer;
};

void __ProfileLoadoutSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __MusicStingerInstanceID__ChangeNotifyFunc();
void __PlayerTitle__ChangeNotifyFunc();
TArray<struct FProductInstanceID> GetAllLoadoutOnlineProducts();
TArray<int32_t> GetAllLoadoutProducts();
void HandleOnlineProductEquipped(class ULoadoutSet_TA* LoadoutSet, struct
FProductInstanceID InstanceID);
void HandleProductEquipped(class ULoadoutSet_TA* LoadoutSet, int32_t ProductID);
void OnEquippedLoadOutChanged();
void HandleEquippedLoadoutChanged(class ULoadoutSet_TA* InLoadoutSet, class
ULoadout_TA* InLoadout);
void HandleLoadoutChanged(class ULoadoutSet_TA* LoadoutSet, class ULoadout_TA* Loadout);
void SetPreviewTeam(int32_t TeamIndex);
void RenamePreset(int32_t Index, class FString NewName);
void EquipLoadoutSet(class ULoadoutSet_TA* InLoadoutSet);
static class FString GetNextPresetName(TArray<class ULoadoutSet_TA*>& InLoadoutSets);
void DeletePreset(int32_t Index);
void CreatePreset(class UProfile_TA* Profile);
void SetLoadoutSets(TArray<class ULoadoutSet_TA*>& InLoadoutSets);
TArray<class ULoadoutSet_TA*> GetLoadoutSets();
class ULoadoutSet_TA* GetLoadoutSet(int32_t SetIndex);
class ULoadout_TA* GetLoadout(int32_t TeamIndex, int32_t SetIndex);
void OnLoad();
void OnCreate();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void EventPreviewTeamChanged(class UProfileLoadoutSave_TA* InLoadoutSave);
void EventOnlineProductEquipped(struct FProductInstanceID InstanceID);
void EventProductEquipped(int32_t ProductID);
void EventEquippedLoadoutChanged(class ULoadout_TA* InLoadout);
};

```

```

// Class TAGame.GFxData_PRI_TA
// 0x0194 (0x0094 - 0x0228)
class UGFxData_PRI_TA : public UGFxDataRow_X
{
public:
    class FString           PlayerName;          // 0x0098 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString           PlayerIDString;      // 0x00A8 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString           Platform;           // 0x00B8 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class UTexture*         Avatar;             // 0x00C8 (0x0008)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Team;                // 0x00D0 (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Score;              // 0x00D4 (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Goals;              // 0x00D8 (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Assists;            // 0x00DC (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Saves;               // 0x00E0 (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Shots;              // 0x00E4 (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 Ping;                // 0x00E8 (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                 BreakoutDamage;     // 0x00EC (0x0004)
    [0x0000000040000000] (CPF>EditInlineNotify)
    unsigned long            bMvp : 1;           // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
    unsigned long            bSpeaking : 1;      // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000002] (CPF>EditInlineNotify)
    unsigned long            bMuted : 1;          // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000004] (CPF>EditInlineNotify)
    unsigned long            bBlocked : 1;        // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000008] (CPF>EditInlineNotify)
    unsigned long            bReady : 1;          // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000010] (CPF>EditInlineNotify)
    unsigned long            bBotPlayer : 1;       // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000020] (CPF>EditInlineNotify)
    unsigned long            bLocalPlayer : 1;     // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000040] (CPF>EditInlineNotify)
    unsigned long            bDisconnected : 1;    // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000080] (CPF>EditInlineNotify)
    unsigned long            bStayAsPartyActive : 1; // 0x00F0 (0x0004)
    [0x0000004040000000] [0x00000100] (CPF>EditInlineNotify)
    unsigned long            bSpectator : 1;        // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000200] (CPF>EditInlineNotify)
    unsigned long            bCompletedPlacementMatches : 1; // 0x00F0
    (0x0004) [0x0000000040000000] [0x00000400] (CPF>EditInlineNotify)
    unsigned long            bSelected : 1;          // 0x00F0 (0x0004)
    [0x0000000040000000] [0x00000800] (CPF>EditInlineNotify)

```

```
unsigned long          bHighlighted : 1;           // 0x00F0 (0x0004)
[0x0000000040000000] [0x00001000] (CPF_EditInlineNotify)
unsigned long          bGameEnded : 1;            // 0x00F0 (0x0004)
[0x0000000000002000] [0x00002000] (CPF_Transient)
unsigned long          bOnPodium : 1;             // 0x00F0 (0x0004)
[0x0000000040000000] [0x00004000] (CPF_EditInlineNotify)
unsigned long          bWantsToHonorDuel : 1;      // 0x00F0 (0x0004)
[0x0001000040000000] [0x00008000] (CPF_EditInlineNotify)
unsigned long          bChallengedToHonorDuel : 1; // 0x00F0 (0x0004)
[0x0001000040000000] [0x00010000] (CPF_EditInlineNotify)
unsigned long          bCanChallengeToHonorDuel : 1; // 0x00F0 (0x0004)
[0x0001000040000000] [0x00020000] (CPF_EditInlineNotify)
class FString          ConnectionStatus;        // 0x00F8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          PartyID;                // 0x0108 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          XPTitle;                // 0x0118 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FColor           TitleColor;              // 0x0128 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FColor           TitleGlowColor;         // 0x012C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                 SkillTier;              // 0x0130 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                 Boost;                  // 0x0134 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class APRI_TA*          PRI;                   // 0x0138 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FUniqueNetId       PlayerID;              // 0x0140 (0x0048)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
float                   PodiumSpotlightScreenPosX; // 0x0188 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float                   PodiumSpotlightScreenPosY; // 0x018C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString           PrimaryTitle;           // 0x0190 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           SecondaryTitle;          // 0x01A0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           PrimaryTitleStat;        // 0x01B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           SecondaryTitleStat;       // 0x01C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                 PrimaryStat;            // 0x01D0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                 SecondaryStat;          // 0x01D4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
uint64_t                ClubID;                 // 0x01D8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t                 WorstConnectionQualityStateBeyondLatency; // 0x01E0
(0x0001) [0x0000000040000000] (CPF_EditInlineNotify)
uint8_t                 QuitSeverity;           // 0x01E1 (0x0001)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
int32_t                 SpectatorShortcut;        // 0x01E4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
```

```

class UGFxData_PlayerBanner_TA*           GFxPlayerBanner;           // 0x01E8
(0x0008) [0x0000000000000000] (CPF_Transient)
class UGFxData_PlayerAvatar_TA*          GFxPlayerAvatar;          // 0x01F0
(0x0008) [0x0000000000000000] (CPF_Transient)
class UGFxData_PlayerAvatarBorder_TA*    GFxPlayerBorder;         // 0x01F8
(0x0008) [0x0000000000000000] (CPF_Transient)
class UPersona_TA*                     Persona;                  // 0x0200 (0x0008)
[0x0000000000000000] (CPF_Transient)
class UPersona_TA*                     PrimaryPersona;          // 0x0208 (0x0008)
[0x0000000000000000] (CPF_Transient)
class UPersonas_TA*                   Personas;                 // 0x0210 (0x0008)
[0x0000800000000000] (CPF_Transient)
class UGFxData_ScoreboardStat_TA*      ScoreboardStats;        // 0x0218
(0x0008) [0x0001000000000000] (CPF_Transient)
class UGFxData_SpectatorStat_TA*      SpectatorStats;         // 0x0220
(0x0008) [0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PRI_TA");
}

return uClassPointer;
};

void __GFxData_PRI_TA__UpdatePRIData_0x4();
int32_t __GFxData_PRI_TA__UpdatePRIData_0x3(struct FScoreboardStat S);
class FString __GFxData_PRI_TA__UpdatePRIData_0x2(struct FScoreboardStat S);
int32_t __GFxData_PRI_TA__UpdatePRIData_0x1(struct FScoreboardStat S);
bool IsActivePlayer();
void HandleQuitSeverityChanged(class APRI_TA* InPRI);
void HandleSpectatorShortcutChanged(class APRI_TA* InPRI);
void HandleReplicatedNetStats();
void HandleClub(class UClubDetails_X* Club);
void HandleClubID();
void UpdatePlayerAvatarBorder();
void UpdatePlayerBanner();
void UpdateScreenPosition(class UCanvas* MyCanvas, struct FVector WorldPosition, unsigned long bHovered, unsigned long bFocused);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void SetPlayerTitle(struct FName TitleId);
void HandleTitleChanged(class APRI_TA* InPRI);
void HandleReplacingBotChanged(class APRI_TA* InPRI);
void InitStatTitle(struct FMemberTitleStat Stat, class FString TitleLocalizationString, class FString& StatTitle, int32_t& StatValue, class FString& StatPostfix);
void HandleStatTitlesSet(class APRI_TA* InPRI);
void HandleWonMVP(class APRI_TA* InPRI);
void HandleSkillTierChanged(class APRI_TA* InPRI);
void HandlePawnTypeChanged(class APRI_TA* InPRI);

```

```

void HandlePartyLeaderChanged(class APRI_TA* InPRI);
void SetDisconnected();
void SetBlocked(unsigned long bInBlocked);
void SetMuted(unsigned long bNewMuted);
void UpdateMuted();
void HandleMuteChanged(class APlayerController_TA* PC, struct FUniqueNetId InPlayerId,
unsigned long bMute);
void UpdatePRIData();
void HandlePlayerTalking(class UOnlineGameVoice_X* SelfRef, struct FUniqueNetId InPlayerId,
unsigned long bTalking);
void HandleReadyChanged(class APRI_TA* InPRI);
void HandleTeamChanged(class APRI_X* InPRI);
void UpdatePlayerAvatar();
void HandleAvatarChange(struct FUniqueNetId InPlayerId);
void HandleVanityUpdate(uint8_t VanityType);
void HandleBlockStatusChanged(class UPersona_TA* InPersona);
void HandlePersonaUpdated(class UPersona_TA* InPersona);
void SetPersonas(class UPersona_TA* InPersona, class UPersona_TA* InPrimaryPersona);
void InitPersonas();
void SetPlayerIDString(class FString Id);
void HandleUniqueIdChanged(class APRI_X* InPRI);
void HandleReceivedLocalPRI(class APlayerController_X* P);
void HandlePlayerNameChanged(class APRI_X* InPRI);
void eventOnRemoved();
void HandleOwnerChanged(class APRI_TA* InPRI);
void SetPRI(class APRI_TA* InPRI);
bool SamePlayer(class APRI_TA* InPRI);
class FString GetReservationStatusString(uint8_t Status);
void SetReservation(struct FReplicatedReservationData Reservation);
class UOnlineGame_X* GetOnlineGame();
};

// Class TAGame.__GFxData_ProductFilter_TA__GetProductAttributeFilters_0x1
// 0x0001 (0x0060 - 0x0061)
class U__GFxData_ProductFilter_TA__GetProductAttributeFilters_0x1 : public UObject
{
public:
    uint8_t FilterType; // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ProductFilter_TA__GetProductAttributeFilters_0x1");
        }

        return uClassPointer;
    };
}

```

```

bool __GFxData_ProductFilter_TA__GetProductAttributeFilters_0x1(struct FFilterProductAttribute
P);
};

// Class TAGame.GFxData_ProductFilter_TA
// 0x0758 (0x0098 - 0x07F0)
class UGFxData_ProductFilter_TA : public UGFxDataSingleton_X
{
public:
TArray<class UClass*> UIAttributes; // 0x0098 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UProductSlot_TA*> AlwaysIgnoredSlots; // 0x00A8
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString Label; // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FFilterProductSlot> Slots; // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FFilterProductQuality> Qualities; // 0x00D8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FFilterProductAttribute> Attributes; // 0x00E8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FFilterProductPaint> Paints; // 0x00F8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FFilterProductSeries> SeriesIDs; // 0x0108 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FFilterProductSorting> SortingLabels; // 0x0118
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
unsigned long bIgnoreEquipped : 1; // 0x0128 (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long bTradeInScreenSelected : 1; // 0x0128 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long bAllowDefaultProducts : 1; // 0x0128 (0x0004)
[0x0000000000004001] [0x00000004] (CPF>Edit | CPF_Config)
float BatchPreviewQuantityTimer; // 0x012C (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<uint8_t> FilterStack; // 0x0130 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FProductFilter Filters[0x4]; // 0x0140 (0x05C0)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class FString> ProductQualitiesToIgnore; // 0x0700 (0x0010)
[0x0000000000404001] (CPF>Edit | CPF_Config | CPF_NeedCtorLink)
TArray<class FString> ProductSortingToIgnore; // 0x0710 (0x0010)
[0x0000000000404001] (CPF>Edit | CPF_Config | CPF_NeedCtorLink)
TArray<class FString> ProductAttributesToIgnore; // 0x0720 (0x0010)
[0x0000000000404001] (CPF>Edit | CPF_Config | CPF_NeedCtorLink)
class FString DefaultSorting; // 0x0730 (0x0010)
[0x0000000000404001] (CPF>Edit | CPF_Config | CPF_NeedCtorLink)
class UGarageArchiveSlot_TA* GarageArchiveSlot; // 0x0740
(0x0008) [0x0000000000002000] (CPF_Transient)
class UGarageFavoritedSlot_TA* GarageFavoritedSlot; // 0x0748
(0x0008) [0x0000000000002000] (CPF_Transient)
class UMtxConfig_TA* MtxConfig; // 0x0750 (0x0008)
[0x0000800000000001] (CPF>Edit)
class UProductsConfig_TA* ProductsConfig; // 0x0758 (0x0008)

```

```

[0x0000800000000000]
struct FScriptDelegate           __EventFilterSelected__Delegate;      // 0x0760
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSlotSelectedChanged__Delegate; // 0x0778
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventAttributeSelected__Delegate; // 0x0790
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPostRefreshPreviewQuantity__Delegate; //
0x07A8 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPreRefreshPreviewQuantity__Delegate; // 0x07C0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventPreFilterProducts__Delegate;     // 0x07D8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductFilter_TA");
}

return uClassPointer;
};

struct FFilterProductSlot __GFxDATA_ProductFilter_TA__InitProductSlots_0x1(class
UProductSlot_TA* Slot);
struct FFilterProductPaint __GFxDATA_ProductFilter_TA__InitProductPaints_0x2(class
UProductPaint_TA* Paint);
bool __GFxDATA_ProductFilter_TA__InitProductPaints_0x1(class UProductPaint_TA* Paint);
class UOnlineProduct_TA* __GFxDATA_ProductFilter_TA__GetTradableCurrencies_0x2(struct
FCurrency C);
bool __GFxDATA_ProductFilter_TA__GetTradableCurrencies_0x1(struct FCurrency C);
int32_t __GFxDATA_ProductFilter_TA__SetupFilterProducts_0x6(struct FProfileProduct P);
bool __GFxDATA_ProductFilter_TA__SetupFilterProducts_0x5(struct FProfileProduct PP);
struct FProductInstanceId __GFxDATA_ProductFilter_TA__SetupFilterProducts_0x4(struct
FProfileProduct P);
bool __GFxDATA_ProductFilter_TA__SetupFilterProducts_0x3(struct FProfileProduct P);
struct FProductInstanceId __GFxDATA_ProductFilter_TA__SetupFilterProducts_0x1(struct
FProfileProduct P);
int32_t __GFxDATA_ProductFilter_TA__FilterProducts_0x1(struct FProductHashID P);
void __GFxDATA_ProductFilter_TA__HandleContainersUpdated_0x1(struct FContainerInfo
Container);
void RemoveNewProduct(struct FProductHashID HashID);
void PopFilterType(uint8_t Type);
void PushFilterType(uint8_t Type);
void ClearFilterType(uint8_t Type);
void UpdateProductSlotNewItem();
void UpdateFromType(uint8_t Type);
void RefreshPreviewQuantity(uint8_t Type);
void HandleBatchPreviewQuantity();
void HandleContainersUpdated(class UGFxDATA.ContainerDrops_TA* ContainerDrops);

```

```
void ApplyBlueprintFilter(uint8_t FilterType, unsigned long bBlueprintSlotSelected);
void ApplyCurrencyTradeRestrictionOption(uint8_t Type, struct FProductFilter& OutFilter);
void ApplyContainerNotificationOption(struct FProductFilter& OutFilter);
void SetStackProducts(unsigned long bValue, uint8_t FilterType);
void SetPackID(struct FProductHashID PackHashID, uint8_t FilterType);
uint8_t GetSortType(uint8_t FilterType);
void SetSortType(uint8_t SortType, uint8_t FilterType);
void SetIgnoreEquipped(unsigned long bSelected, uint8_t FilterType);
void SetPaintSelected(int32_t InPaintID, unsigned long bSelected, uint8_t FilterType);
void SetSeriesIDSelected(int32_t Row, unsigned long bSelected, uint8_t FilterType);
void SetProhibitedAttributeTypeSelected(uint8_t Attribute, unsigned long bSelected, uint8_t FilterType);
void SetExclusiveAttributeTypeSelected(uint8_t Attribute, unsigned long bSelected, uint8_t FilterType);
static struct FName FilterProductAttributeToName(uint8_t Attribute);
void HandleAttributeSelected(unsigned long bSelected, uint8_t FilterType, class UClass* CurrentAttributeClass);
void HandleDummyAttributeSelected(unsigned long bSelected, uint8_t FilterType, class UClass* CurrentAttributeClass);
void SetAttributeSelected(class FString AttributeLabel, unsigned long bSelected, uint8_t FilterType);
void RemoveProhibitedQuality(uint8_t Quality, uint8_t FilterType);
void AddProhibitedQuality(uint8_t Quality, uint8_t FilterType);
void SetTradeInSelected(unsigned long bSelected);
TArray<struct FFilterProductAttribute> GetProductAttributeFilters(uint8_t FilterType);
TArray<struct FFilterProductSorting> GetProductSortingFilters(uint8_t FilterType);
void OnTradeInFilterSelected(uint8_t FilterType);
void SetTradeInFilterSelected(int32_t TradeInFilterID, uint8_t Quality, uint8_t FilterType);
void SetQualitySelected(int32_t Row, unsigned long bSelected, uint8_t FilterType);
void SetSlotSelected(int32_t Row, unsigned long bSelected, uint8_t FilterType);
void SetLabel(class FString NewLabel, uint8_t FilterType);
TArray<struct FProductHashID> FilterProducts(uint8_t FilterType);
void SetupFilterProducts(uint8_t FilterType, unsigned long bIncludeDuplicateProducts);
TArray<class UOnlineProduct_TA*> GetTradableCurrencies();
bool ShouldDisplayOfflineProducts(uint8_t FilterType);
void UpdateProductPaints(struct FProductFilter& InFilter);
void UpdateProductAttributes(struct FProductFilter& InFilter);
void UpdateProductQualities(struct FProductFilter& InFilter);
void UpdateProductSlots(uint8_t Type);
int32_t GetFilterQuantity(struct FProductFilter& InFilter);
void InitSortTypes();
void AddUniqueSeriesID(int32_t SeriesID);
void InitProductPaints();
void InitProductSeriesIDs();
void InitProductAttributes();
void InitProductQualities();
void InitProductSlots();
void HandleNewProductUnlocked(class USaveData_TA* Data, class UOnlineProduct_TA* OnlineProduct, class FString Message);
void InitFromProfile(class ULocalPlayer_TA* LocalPlayer);
void eventOnShellSet();
class UProfile_TA* GetProfile();
void EventPreFilterProducts(uint8_t SelectedFilter);
void EventPreRefreshPreviewQuantity(uint8_t SelectedFilter);
```

```

void EventPostRefreshPreviewQuantity(uint8_t SelectedFilter);
void EventAttributeSelected(class UClass* SelectedAttributeClass, uint8_t SelectedFilter);
void EventSlotSelectedChanged(int32_t SlotIndex, unsigned long bSelected, uint8_t
SelectedFilter);
void EventFilterSelected(uint8_t SelectedFilter);
};

// Class TAGame.__GFxData_ProductFilter_TA__GetProductSortingFilters_0x1
// 0x0001 (0x0060 - 0x0061)
class U__GFxData_ProductFilter_TA__GetProductSortingFilters_0x1 : public UObject
{
public:
    uint8_t FilterType; // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ProductFilter_TA__GetProductSortingFilters_0x1");
        }
    }

    return uClassPointer;
};

bool __GFxData_ProductFilter_TA__GetProductSortingFilters_0x1(struct FFilterProductSorting P);
};

// Class TAGame.__GFxData_ProductFilter_TA__InitProductAttributes_0x1
// 0x0010 (0x0060 - 0x0070)
class U__GFxData_ProductFilter_TA__InitProductAttributes_0x1 : public UObject
{
public:
    TArray<class UClass*> ExcludedAttributes; // 0x0060 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ProductFilter_TA__InitProductAttributes_0x1");
        }
    }

    return uClassPointer;
};

```

```
void __GFxData_ProductFilter_TA_InitProductAttributes_0x1(class FString IgnoreAttribute);
};

// Class TAGame.__GFxData_ProductFilter_TA_SetExclusiveAttributeTypeSelected_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_ProductFilter_TA_SetExclusiveAttributeTypeSelected_0x1 : public UObject
{
public:
    struct FName           Typename;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ProductFilter_TA_SetExclusiveAttributeTypeSelected_0x1");
        }

        return uClassPointer;
    }

    bool __GFxData_ProductFilter_TA_SetExclusiveAttributeTypeSelected_0x1(class UClass* A);
};

// Class TAGame.__GFxData_ProductFilter_TA_SetProhibitedAttributeTypeSelected_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_ProductFilter_TA_SetProhibitedAttributeTypeSelected_0x1 : public UObject
{
public:
    struct FName           Typename;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ProductFilter_TA_SetProhibitedAttributeTypeSelected_0x1");
        }

        return uClassPointer;
    }

    bool __GFxData_ProductFilter_TA_SetProhibitedAttributeTypeSelected_0x1(class UClass* A);
};

// Class TAGame.__GFxData_ProductFilter_TA_SetupFilterProducts_0x2
```

```

// 0x0028 (0x0060 - 0x0088)
class U_GFxDATA_ProductFilter_TA_SetupFilterProducts_0x2 : public UObject
{
public:
class USaveData_TA*           SaveData;          // 0x0060 (0x0008)
[0x0000000000000000]
TArray<int32_t>             AllEquippedProducts; // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceID> AllEquippedOnlineProducts; // 0x0078
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_ProductFilter_TA_SetupFilterProducts_0x2");
}

return uClassPointer;
};

bool __GFxDATA_ProductFilter_TA_SetupFilterProducts_0x8(class UOnlineProduct_TA* Op);
bool __GFxDATA_ProductFilter_TA_SetupFilterProducts_0x7(class UProduct_TA* P);
struct FProductHashID __GFxDATA_ProductFilter_TA_SetupFilterProducts_0x2(struct
FProductInstanceID P);
};

// Class TAGame._GFxDATA_ProductFilter_TA_UpdateProductSlotNewItems_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxDATA_ProductFilter_TA_UpdateProductSlotNewItems_0x1 : public UObject
{
public:
int32_t                      SlotIdx;          // 0x0060 (0x0004)
[0x0000000000000000]
class ULoadout_TA*           Loadout;          // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_ProductFilter_TA_UpdateProductSlotNewItems_0x1");
}

return uClassPointer;
};

```

```

bool __GFxData_ProductFilter_TA__UpdateProductSlotNewItems_0x2(class UProduct_TA* Product);
bool __GFxData_ProductFilter_TA__UpdateProductSlotNewItems_0x1(class UProduct_TA* Product);
};

// Class TAGame.Loadout_TA
// 0x00D8 (0x0060 - 0x0138)
class ULoadout_TA : public UObject
{
public:
TArray<int32_t> Products; // 0x0060 (0x0010)
[0x0000008000400000] (CPF_NeedCtorLink)
TArray<uint64_t> OnlineProducts; // 0x0070 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceID> OnlineProducts128; // 0x0080
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FLoadoutTeamPaint TeamPaint; // 0x0090 (0x0010)
[0x0000008000000000]
struct FLoadoutTeamPaint TeamPaints[0x2]; // 0x00A0 (0x0020)
[0x0000004000000002] (CPF_Const)
struct FName PlayerTitle; // 0x00C0 (0x0008)
[0x0000004000000000]
int32_t GaragePreviewTeam; // 0x00C8 (0x0004)
[0x0000004000000000]
class UProfile_TA* Profile; // 0x00D0 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FScriptDelegate __EventChanged__Delegate; // 0x00D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPreviewTeamChanged__Delegate; // 0x00F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventProductEquipped__Delegate; // 0x0108
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventOnlineProductEquipped__Delegate; // 0x0120
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Loadout_TA");
}

return uClassPointer;
};

static bool __Loadout_TA__ApplyCompatibleAttributeProducts_0x2(class UProduct_TA* B);
void PrintDebugInfo(class UDebugDrawer* Drawer);
class UProductAsset_Body_TA* LoadBodyAsset();
int32_t GetBodyProductID();

```

```

bool IsProductEquipped(class UProductSlot_TA* Slot, int32_t ProductID);
void SetGaragePreviewTeam(int32_t PreviewTeam);
void SetDataForced(struct FLoadoutData& InLoadout);
void SetData(struct FLoadoutData& Data);
struct FLoadoutData GetData();
static void ApplyCompatibleAttributeProducts(int32_t ProductID, struct FLoadoutData& Data);
static void ApplyRequiredProducts(int32_t ProductID, struct FLoadoutData& Data);
static void ValidateForcedProducts(int32_t TeamIndex, struct FLoadoutData& Data);
static bool IsCustomFinishDisabled(class UProductAsset_Body_TA* BodyAsset, class
UProductAsset_Skin_TA* SkinAsset);
static bool IsTeamFinishDisabled(class UProductAsset_Body_TA* BodyAsset, class
UProductAsset_Skin_TA* SkinAsset);
static void SetBodyProduct(int32_t BodyProductID, struct FLoadoutData& OutLoadout);
static void ValidateProductID(class UProfile_TA* ForProfile, class FString SlotName, int32_t&
ProductID);
static bool IsForcedSlot(int32_t BodyProductID, class UProductSlot_TA* Slot);
static TArray<class UProduct_TA*> GetForcedProducts(class UProductAsset_Body_TA* InBody);
static void StaticValidate(class UProfile_TA* ForProfile, struct FLoadoutTeamPaint& Paint,
TArray<int32_t>& OutProducts, TArray<struct FProductInstanceID>& OutOnlineProducts);
void Validate();
static void SetBodyProductAndValidate(int32_t BodyProductID, class UProfile_TA* ForProfile,
struct FLoadoutData& OutLoadout);
static struct FLoadoutData EquipProduct(struct FProductInstanceID OnlineID, class
ULoadout_TA* ForLoadout, int32_t SlotIndex);
void InitProfile(class UProfile_TA* InProfile);
void Init(class UProfile_TA* InProfile, int32_t TeamIndex);
void EventOnlineProductEquipped(struct FProductInstanceID InstanceID);
void EventProductEquipped(int32_t ProductID);
void EventPreviewTeamChanged(class ULoadout_TA* Loadout);
void EventChanged(class ULoadout_TA* Loadout);
};

// Class TAGame.GFxData_Products_TA
// 0x0170 (0x0098 - 0x0208)
class UGFxData_Products_TA : public UGFxDataSingleton_X
{
public:
TArray<class ULoadingProduct_TA*> LoadingProducts; // 0x0098
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FProductStatInfo> Stats; // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ProductLabel; // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ProductLongLabel; // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture* ProductIcon; // 0x00D8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString ProductDescription; // 0x00E0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FProductInstanceID ProductID; // 0x00F0 (0x0010)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t BaseProductID; // 0x0100 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FProductHashID HashID; // 0x0104 (0x0004)

```

```

[0x0000000040000000] (CPF_EditInlineNotify)
struct FProductHashID           ProductPackHashID;          // 0x0108 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                         SchematicPointerID;        // 0x010C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                         ProductSlot;              // 0x0110 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString                   ProductSlotLabel;         // 0x0118 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString                   ProductSlotLongLabel;      // 0x0128 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
unsigned long                   bProductEquipped : 1;    // 0x0138 (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long                   bProductEquipped01 : 1;   // 0x0138 (0x0004)
[0x0000000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long                   bProductUnlocked : 1;     // 0x0138 (0x0004)
[0x0000000040000000] [0x00000004] (CPF>EditInlineNotify)
unsigned long                   bProductIsPack : 1;       // 0x0138 (0x0004)
[0x0000000040000000] [0x00000008] (CPF>EditInlineNotify)
unsigned long                   bProductIsNew : 1;        // 0x0138 (0x0004)
[0x0000000040000000] [0x00000010] (CPF>EditInlineNotify)
unsigned long                   bProductHasNewSkin : 1;   // 0x0138 (0x0004)
[0x0000000040000000] [0x00000020] (CPF>EditInlineNotify)
unsigned long                   bProductCanBeTradedIn : 1; // 0x0138 (0x0004)
[0x0000000040000000] [0x00000040] (CPF>EditInlineNotify)
unsigned long                   bProductCanBePlayerTraded : 1; // 0x0138 (0x0004)
[0x0000000040000000] [0x00000080] (CPF>EditInlineNotify)
unsigned long                   bProductIsContainer : 1;    // 0x0138 (0x0004)
[0x0000000040000000] [0x00000100] (CPF>EditInlineNotify)
unsigned long                   bProductIsContainerKey : 1; // 0x0138 (0x0004)
[0x0000000040000000] [0x00000200] (CPF>EditInlineNotify)
unsigned long                   bProductIsSchematic : 1;    // 0x0138 (0x0004)
[0x0000000040000000] [0x00000400] (CPF>EditInlineNotify)
unsigned long                   bHasPostMatchCelebration : 1; // 0x0138 (0x0004)
[0x0000000040000000] [0x00000800] (CPF>EditInlineNotify)
unsigned long                   bIsReactive : 1;          // 0x0138 (0x0004)
[0x0000000040000000] [0x00001000] (CPF>EditInlineNotify)
unsigned long                   bHasReactivePreview : 1;   // 0x0138 (0x0004)
[0x0000000040000000] [0x00002000] (CPF>EditInlineNotify)
unsigned long                   bIsFavorited : 1;        // 0x0138 (0x0004)
[0x0000000040000000] [0x00004000] (CPF>EditInlineNotify)
unsigned long                   bIsArchived : 1;        // 0x0138 (0x0004)
[0x0000000040000000] [0x00008000] (CPF>EditInlineNotify)
unsigned long                   bCanArchive : 1;        // 0x0138 (0x0004)
[0x0000000040000000] [0x00010000] (CPF>EditInlineNotify)
unsigned long                   bIsXETradeLocked : 1;    // 0x0138 (0x0004)
[0x0001000040000000] [0x00020000] (CPF>EditInlineNotify)
unsigned long                   bDebugLoadingProducts : 1; // 0x0138 (0x0004)
[0x0000000000000001] [0x00040000] (CPF>Edit)
struct FColor                  ProductPaintColor;          // 0x013C (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString                  ProductPaintColorName;      // 0x0140 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
uint8_t                         ProductQuality;           // 0x0150 (0x0001)

```

```

[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t XEStatus; // 0x0151 (0x0001)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t ProductQuantity; // 0x0154 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t TradeHold; // 0x0158 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t SeriesID; // 0x015C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString SeriesDescription; // 0x0160 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString XEShopDescriptionText; // 0x0170 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString XEBLueprintDescriptionText; // 0x0180 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString XEDescriptionText; // 0x0190 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t ProductTradeInQuantity; // 0x01A0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t SchematicResources; // 0x01A4 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class UGFxProductAttributeMap_TA* AttributeMap; // 0x01A8
(0x0008) [0x000000004400009] (CPF>Edit | CPF>ExportObject | CPF>NeedCtorLink |
CPF>EditInline)
class FString AttributeTables; // 0x01B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<class UProduct_TA*> NewProducts; // 0x01C0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UProductsFavoriteSave_TA* FavoriteSave; // 0x01D0
(0x0008) [0x0000000000002000] (CPF>Transient)
class UProductsArchiveSave_TA* ArchiveSave; // 0x01D8
(0x0008) [0x0000000000002000] (CPF>Transient)
class UProfileLoadoutSave_TA* LoadoutSave; // 0x01E0 (0x0008)
[0x0000000000002000] (CPF>Transient)
TArray<class UClass*> DynamicThumbnailsIgnorableOnlinePaintedProducts;//
0x01E8 (0x0010) [0x0000000000402000] (CPF>Transient | CPF>NeedCtorLink)
class UMtxConfig_TA* MtxConfig; // 0x01F8 (0x0008)
[0x0000800000000001] (CPF>Edit)
class UProductsConfig_TA* ProductsConfig; // 0x0200 (0x0008)
[0x0000800000000001] (CPF>Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Products_TA");
}

return uClassPointer;
};

```

```
void __GFxData_Products_TA__OnShellSet_0x4(class UWallet_TA* Wallet);
void __GFxData_Products_TA__OnShellSet_0x5(class UWallet_TA* _);
void __GFxData_Products_TA__OnShellSet_0x3(class UProductsArchiveSave_TA*
InArchiveSave);
void __GFxData_Products_TA__OnShellSet_0x2(class UProfileLoadoutSave_TA* InLoadoutSave);
void __GFxData_Products_TA__OnShellSet_0x1(class UProductsFavoriteSave_TA*
InFavoriteSave);
bool __GFxData_Products_TA__GetAttributesWithAdditionalAssets_0x1(class
UProductAttribute_TA* A);
class UGFxData_ProductAttribute_TA* __GFxData_Products_TA__InstanceAttributes_0x2(class
UClass* GFxClass);
class UClass* __GFxData_Products_TA__InstanceAttributes_0x1(class UProductAttribute_TA*
Attribute);
struct FName __GFxData_Products_TA__BuildAttributeTableStr_0x1(class
UGFxData_ProductAttribute_TA* GFxAttribute);
bool __GFxData_Products_TA__HasNewOnlineProducts_0x1(struct FProductInstanceID Id);
bool __GFxData_Products_TA__FolderContainsNewProducts_0x1(class UProduct_TA* P);
struct FProductInstanceID
__GFxData_Products_TA__GetMatchingInstancesByProductID_0x1(class UOnlineProduct_TA*
OnlineProduct);
class ACarPreviewActor_TA* GetCarPreview();
int32_t GetEditTeamIndex(class UProfile_TA* Profile);
class ULoadoutSet_TA* GetEditLoadoutSet(class UProfile_TA* Profile);
class ULoadout_TA* GetEditLoadout(class UProfile_TA* Profile, int32_t TeamIndex);
void OnSchematicResourcesChanged();
void HandleSchematicResourcesChanged(class USaveData_TA* SaveData);
void RefreshSchematicResources();
void HandleRemovedOnlineProduct(class USaveData_TA* Data, class UOnlineProduct_TA*
OnlineProduct);
void HandleNewOnlineProduct(class USaveData_TA* Data, class UOnlineProduct_TA*
OnlineProduct, class FString Message);
void OnProductsRefreshed();
void HandleOnlineProductsSynced(class USaveData_TA* SaveData);
void TimerRefreshProducts();
void RefreshProducts();
TArray<struct FProductInstanceID> GetMatchingInstancesByProductID(int32_t InProductID);
int32_t GetTimeHoldRemaining(struct FProductHashID InHashID);
void UpdateTeamProductData();
void UpdateAllProductData();
struct FProductHashID GetNewestProductId();
bool SlotContainsNewProducts(int32_t SlotIndex);
bool FolderContainsNewProducts();
bool HasNewOnlineProducts();
class FString GetProductLongLabel(struct FProductHashID InHashID);
uint8_t GetProductQuality(struct FProductHashID InHashID);
class FString GetProductTrademarkLabel(struct FProductHashID InHashID);
class FString GetProductLabel(struct FProductHashID InHashID);
bool BaseProductsMatch(struct FProductHashID FirstProduct, struct FProductHashID
SecondProduct);
struct FProductHashID GetHashIDFromProductID(int32_t InProductID);
class FString GetSlotLabel(struct FProductHashID InHashID);
int32_t GetSlotIndex(struct FProductHashID InHashID);
bool IsContainer(struct FProductHashID InHashID);
void OnProductFailedToLoad(struct FProductHashID InHashID);
```

```
void OnProductLoaded(struct FProductHashID InHashID, int32_t ThumbnailSize);
class FString BuildAttributeTableStr(TArray<class UGFxData_ProductAttribute_TA*>& Attributes);
TArray<class UGFxData_ProductAttribute_TA*> InstanceAttributes(class
UIProductAttributeProvider_TA* AttributeProvider);
void ClearAttributes();
void UpdateProductData(class ULoadingProduct_TA* ProductData);
bool UpdateEquippedVanity(int32_t SlotIndex, struct FProductHashID ProductHash);
void UpdateNextProductData();
void SetUpdateProductDataTimer();
void QueReRenderOfProduct(class ULoadingProduct_TA* LoadingProduct);
void QueUpdateProductData(class ULoadingProduct_TA* LoadingProduct);
bool CanBePlayerTraded(class UProduct_TA* Product);
bool CanBeTradedIn(class UProduct_TA* Product);
void HandleRenderTargetUpdate(class UTextureRenderTarget2D* Texture, class
ULoadingProduct_TA* LoadingProduct);
void HandleThumbnailLoaded(struct FAssetLoadResult Result, class ULoadingProduct_TA*
LoadingProduct);
void HandleProductThumbnailRendered(struct FProductThumbnailResult Result, class
ULoadingProduct_TA* LoadingProduct);
void RenderProductThumbnail(class ULoadingProduct_TA* ProductData);
void HandleAdditionalProductReferenceLoaded(class ULoadingProduct_TA* LoadingProduct,
struct FAssetLoadResult& Result);
void HandlePaintedProductLoaded(class ULoadingProduct_TA* LoadingProduct, struct
FAssetLoadResult& Result);
TArray<class UProductAttribute_TA*> GetAttributesWithAdditionalAssets(class
ULoadingProduct_TA* ProductData);
void LoadAdditionalProductReferences(class ULoadingProduct_TA* LoadingProduct);
void SetIsArchived(struct FProductHashID InHashID, unsigned long bValue);
void SetIsFavorited(struct FProductHashID InHashID, unsigned long bValue);
void StartLoadingProduct(struct FProductHashID InHashID, uint8_t ThumbnailSize, int32_t
ViewId, unsigned long bIgnoreEquipped);
class UOnlineProductStoreSet_TA* GetLocalOnlineProductSet();
void RefreshProduct(class ULoadingProduct_TA* LoadingProduct, int32_t ViewId, unsigned long
bIgnoreEquipped);
void LoadProduct(struct FProductHashID InHashID, uint8_t ThumbnailSize, int32_t ViewId,
unsigned long bIgnoreEquipped);
void OnClearViewedProduct(struct FProductHashID InHashID, uint8_t ThumbnailSize);
bool IsThumbnailReferenced(class UProductAsset_TA* Asset, class UOnlineProduct_TA*
OnlineProduct, uint8_t ThumbnailSize);
bool IsProductIDReferenced(int32_t InProductID);
void ClearProductView(class ULoadingProduct_TA* LoadingProduct, int32_t ViewId);
void ClearReferencedProducts(int32_t ViewId);
void SetViewProducts(int32_t ViewId, TArray<struct FProductHashID> HashIDs, uint8_t
ThumbnailSize);
TArray<struct FProductInstanceID> GetVisibleProductIDs(int32_t SlotIndex, struct
FProductHashID PackHashID, unsigned long bUnlockedOnly);
TArray<struct FProductInstanceID> GetUnlockedProductIDs(int32_t SlotIndex);
static bool ShouldShowProduct(class ULoadout_TA* Loadout, class UProduct_TA* Product,
struct FProductHashID PackHashID);
void CacheNewProducts();
void HandleProductsSave(class UProductsSave_TA* ProductsSave);
class UProfile_TA* GetProfile();
void eventOnRemoved();
void eventOnShellSet();
```

```

};

// Class TAGame.__GFxData_Products_TA_HandlePaintedProductLoaded_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_Products_TA_HandlePaintedProductLoaded_0x1 : public UObject
{
public:
class ULoadingProduct_TA*           LoadingProduct;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_Products_TA_HandlePaintedProductLoaded_0x1");
}

return uClassPointer;
};

void __GFxDATA_Products_TA_HandlePaintedProductLoaded_0x1(struct FAssetLoadResult
AssetResult);
};

// Class TAGame.LoadingProduct_TA
// 0x0058 (0x0060 - 0x00B8)
class ULoadingProduct_TA : public UObject
{
public:
int32_t                           ProductID;             // 0x0060 (0x0004)
[0x0000000000000000]
struct FProductHashID              HashID;                // 0x0064 (0x0004)
[0x0000000000000000]
TArray<int32_t>                   ReferencedViews;      // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UOnlineProduct_TA*            OnlineProduct;        // 0x0078 (0x0008)
[0x0000000000000000]
uint8_t                            ThumbnailSize;        // 0x0080 (0x0001)
[0x0000000000000000]
class UProductThumbnailAsset_TA*    ThumbnailAsset;        // 0x0088
(0x0008) [0x0000000000000000]
class UProductAsset_TA*             Asset;                 // 0x0090 (0x0008)
[0x0000000000000000]
class UProductThumbnail_TA*         Thumbnail;             // 0x0098 (0x0008)
[0x0000000000000000]
unsigned long                      bUpdateProductData : 1; // 0x00A0 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long                      bIgnoreEquipped : 1;   // 0x00A0 (0x0004)
[0x0000000000000000] [0x00000002]
TArray<class UObject*>            AdditionalProductReferences; // 0x00A8

```

(0x0010) [0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.LoadingProduct_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.__GFxData_Products_TA__HandleProductThumbnailRendered_0x1  
// 0x0008 (0x0060 - 0x0068)  
class U__GFxData_Products_TA__HandleProductThumbnailRendered_0x1 : public UObject  
{  
public:  
class ULoadingProduct_TA*           LoadingProduct;          // 0x0060 (0x0008)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__GFxData_Products_TA__HandleProductThumbnailRendered_0x1");  
}  
  
return uClassPointer;  
};  
  
void __GFxData_Products_TA__HandleProductThumbnailRendered_0x1(class  
UTextureRenderTarget2D* ThumbnailTexture);  
};  
  
// Class TAGame.__GFxData_Products_TA__InstanceAttributes_0x3  
// 0x0010 (0x0060 - 0x0070)  
class U__GFxData_Products_TA__InstanceAttributes_0x3 : public UObject  
{  
public:  
class UIProductAttributeProvider_TA*   AttributeProvider_Object;    // 0x0060  
(0x0008) [0x0000000000000000]  
class UIProductAttributeProvider_TA*   AttributeProvider_Interface; // 0x0068  
(0x0008) [0x0000000000000000]  
  
public:
```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Products_TA__InstanceAttributes_0x3");
}

return uClassPointer;
};

void __GFxData_Products_TA__InstanceAttributes_0x3(class UGFxData_ProductAttribute_TA*
GFxAttribute);
};

// Class TAGame.IProductAttributeProvider_TA
// 0x0000 (0x0060 - 0x0060)
class UIProductAttributeProvider_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IProductAttributeProvider_TA");
}

return uClassPointer;
};

class UProductAttribute_TA* GetAttribute(class UClass* AttributeClass);
TArray<class UProductAttribute_TA*> GetAttributes();
};

// Class TAGame.GFxData_ProductAttribute_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_ProductAttribute_TA : public UGFxDataSingleton_X
{
public:
class UClass* AttributeClass; // 0x0098 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UIProductAttributeProvider_TA* AttributeProvider_Object; // 0x00A0
(0x0008) [0x0000004000002000] (CPF_Transient)
class UIProductAttributeProvider_TA* AttributeProvider_Interface; // 0x00A8
(0x0008) [0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_TA");
}

return uClassPointer;
};

void UpdateAdditionalLoadedData(class UGFxData_Products_TA* GFxData_Products, class
ULoadingProduct_TA* ProductData, class UProduct_TA* Product);
void OnReceivedAttribute(class UProductAttribute_TA* Attribute);
void Init(class UIProductAttributeProvider_TA* InProvider);
};

// Class TAGame.__GFxData_Products_TA__IsThumbnailReferenced_0x1
// 0x0011 (0x0060 - 0x0071)
class U__GFxData_Products_TA__IsThumbnailReferenced_0x1 : public UObject
{
public:
class UProductAsset_TA*           Asset;                      // 0x0060 (0x0008)
[0x0000000000000000]
class UOnlineProduct_TA*          OnlineProduct;        // 0x0068 (0x0008)
[0x0000000000000000]
uint8_t                           ThumbnailSize;       // 0x0070 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Products_TA__IsThumbnailReferenced_0x1");
}

return uClassPointer;
};

bool __GFxData_Products_TA__IsThumbnailReferenced_0x1(class ULoadingProduct_TA* L);
};

// Class TAGame.__GFxData_Products_TA__LoadAdditionalProductReferences_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Products_TA__LoadAdditionalProductReferences_0x1 : public UObject
{
public:
class ULoadingProduct_TA*          LoadingProduct;      // 0x0060 (0x0008)
[0x0000000000000000]
}

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Products_TA_LoadAdditionalProductReferences_0x1");
}

return uClassPointer;
};

void
__GFxData_Products_TA_LoadAdditionalProductReferences_0x1__GFxData_Products_TA_Lo
adAdditionalProductReferences_0x2_0x1(struct FAssetLoadResult Result);
void __GFxData_Products_TA_LoadAdditionalProductReferences_0x2(class
UProductAttribute_TA* A);
void __GFxData_Products_TA_LoadAdditionalProductReferences_0x3(struct FAssetLoadResult
Result);
void __GFxData_Products_TA_LoadAdditionalProductReferences_0x1(struct FAssetLoadResult
AssetResult);
};

// Class TAGame._GFxData_Products_TA_LoadProduct_0x1
// 0x0005 (0x0060 - 0x0065)
class U_GFxData_Products_TA_LoadProduct_0x1 : public UObject
{
public:
struct FProductHashID           InHashID;          // 0x0060 (0x0004)
[0x0000000000000000]
uint8_t                          ThumbnailSize;    // 0x0064 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Products_TA_LoadProduct_0x1");
}

return uClassPointer;
};

bool __GFxData_Products_TA_LoadProduct_0x1(class ULoadingProduct_TA* P);
};

// Class TAGame._GFxData_Products_TA_RenderProductThumbnail_0x1
// 0x0008 (0x0060 - 0x0068)

```

```
class U_GFxDATA_Products_TA_RenderProductThumbnail_0x1 : public UObject
{
public:
class ULoadingProduct_TA*          ProductData;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_Products_TA_RenderProductThumbnail_0x1");
}

return uClassPointer;
};

void _GFxDATA_Products_TA_RenderProductThumbnail_0x1(struct FProductThumbnailResult
ThumbnailResult);
};

// Class TAGame._GFxDATA_Products_TA_SlotContainsNewProducts_0x1
// 0x0004 (0x0060 - 0x0064)
class U_GFxDATA_Products_TA_SlotContainsNewProducts_0x1 : public UObject
{
public:
int32_t                SlotIndex;           // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_Products_TA_SlotContainsNewProducts_0x1");
}

return uClassPointer;
};

bool _GFxDATA_Products_TA_SlotContainsNewProducts_0x1(class UProduct_TA* P);
};

// Class TAGame._GFxDATA_Products_TA_StartLoadingProduct_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_Products_TA_StartLoadingProduct_0x1 : public UObject
{
public:
```

```

class ULoadingProduct_TA*           ProductToLoad;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_Products_TA_StartLoadingProduct_0x1");
}

return uClassPointer;
};

void __GFxDATA_Products_TA_StartLoadingProduct_0x2(struct FAssetLoadResult AssetResult);
void __GFxDATA_Products_TA_StartLoadingProduct_0x1(struct FAssetLoadResult
PaintedAssetResult);
};

// Class TAGame._GFxDATA_Products_TA_UpdateProductData_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxDATA_Products_TA_UpdateProductData_0x1 : public UObject
{
public:
class ULoadingProduct_TA*           ProductData;          // 0x0060 (0x0008)
[0x0000000000000000]
class UProduct_TA*                 Product;              // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_Products_TA_UpdateProductData_0x1");
}

return uClassPointer;
};

bool __GFxDATA_Products_TA_UpdateProductData_0x3(class UProduct_TA* P);
void __GFxDATA_Products_TA_UpdateProductData_0x2(class UGFxDATA_ProductAttribute_TA*
GFxAttribute);
bool __GFxDATA_Products_TA_UpdateProductData_0x1(class UProductAttribute_TA* A);
};

// Class TAGame._GFxDATA_ProductTradeIn_TA_UpdateTradeInQuantity_0x1
// 0x0004 (0x0060 - 0x0064)

```

```

class U_GFxDATA_ProductTradeIn_TA__UpdateTradeInQuantity_0x1 : public UObject
{
public:
    struct FProductHashID           HashID;           // 0x0060 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_ProductTradeIn_TA__UpdateTradeInQuantity_0x1");
        }

        return uClassPointer;
    };

    bool __GFxDATA_ProductTradeIn_TA__UpdateTradeInQuantity_0x2(struct FTradeIn Trade);
    bool __GFxDATA_ProductTradeIn_TA__UpdateTradeInQuantity_0x1(struct FTradeIn Trade);
};

// Class TAGame.GFxDATA_ProductTradeIn_TA
// 0x0068 (0x0098 - 0x0100)
class UGFxDATA_ProductTradeIn_TA : public UGFxDATA_Singleton_X
{
public:
    TArray<struct FTradeIn>           TradeIns;          // 0x0098 (0x0010)
    [0x0000004040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class UProductTransactions_TA*     ProductTransactions; // 0x00A8
    (0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
    CPF_EditInline)
    class UGFxModal_X*                ModalProcessing;   // 0x00B0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class FString                     InvalidProductString; // 0x00B8 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                     NotEnoughProductsSelected; // 0x00C8 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    int32_t                           NumProductsRequiredForTradeIn; // 0x00D8 (0x0004)
    [0x0000000000000002] (CPF_Const)
    class UProductsConfig_TA*         ProductsConfig;    // 0x00E0 (0x0008)
    [0x0001800000000000]
    struct FScriptDelegate             __EventTradeInComplete__Delegate; // 0x00E8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {

```

```

uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductTradeIn_TA");
}

return uClassPointer;
};

bool __GFxData_ProductTradeIn_TA__HasBlueprintTradeIns_0x1(class UOnlineProduct_TA* P);
bool HasBlueprintTradeIns();
class UProfile_TA* GetProfile();
void OnTradeInFailed();
void OnTradeInComplete();
void HandleTradeInComplete(class URPC_ProductsTradeIn_TA* RPC);
void PerformTradeIn();
void ClearTradeIns();
void CancelTradeIn();
void UpdateTradeInQuantity(struct FProductHashID HashID, int32_t DesiredQuantity);
void SetTransactionQuantity(struct FProductHashID HashID, int32_t Quantity);
bool IsProcessingTrade();
void HandleProductsConfigChanged();
void HandleProfileSet(class ULocalPlayer_TA* InLocalPlayer);
void eventOnRemoved();
void eventOnShellSet();
void EventTradeInComplete(class UGFxData_ProductTradeIn_TA* GFxData, TArray<class UOnlineProduct_TA*> Given, TArray<class UOnlineProduct_TA*> Received);
};

// Class TAGame.__GFxData_ReplayManager_TA__DownloadReplay_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_ReplayManager_TA__DownloadReplay_0x1 : public UObject
{
public:
class UWebRequest_X* Request; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ReplayManager_TA__DownloadReplay_0x1");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ReplayManager_TA
// 0x0058 (0x0098 - 0x00F0)
class UGFxData_ReplayManager_TA : public UGFxDataSingleton_X
{

```

```

public:
class FString           ReplaySaveID;           // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           ReplaySaveError;        // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<class UGFxData_Replay_TA*>   ReplayData;       // 0x00B8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FReplayHeaderLoadResult> ReplayLoadResults; // 0x00C8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UGFxModal_X*      ModalProcessing;        // 0x00D8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString            PendingSaveName;         // 0x00E0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ReplayManager_TA");
}

return uClassPointer;
};

int32_t __GFxDatA_ReplayManager_TA__HandleAlternateHeadersLoaded_0x1(struct
FReplayHeaderLoadResult Left, struct FReplayHeaderLoadResult Right);
void PrintReplays();
void HandleReplayImported(class UReplayManager_TA* Manager, class UReplay_TA* InReplay,
class UError* Error);
void RenameReplay(class FString Id, class FString NewReplayName);
void DeleteReplay(class FString Id);
void StartReplayID(class FString Id);
int32_t GetReplayIndexFromID(class FString Id);
void StartReplayIndex(int32_t Index);
void OnReplaysLoaded();
void UpdateReplayData(TArray<struct FReplayHeaderLoadResult> Headers);
void HandleAlternateHeadersLoaded(class UReplayManager_TA* Manager, TArray<struct
FReplayHeaderLoadResult> Headers, class UError* Error);
void HandleHeadersLoaded(class UReplayManager_TA* Manager, TArray<struct
FReplayHeaderLoadResult> Headers, class UError* Error);
void MarkReplayEditorOpened();
void LoadReplays();
void OnReplayExportFailed();
void OnReplayExportSuccess();
bool IsValidReplayFilename(class FString Filename);
bool IsReplayFilenameTaken(class FString Filename);
void HandleReplayExported(class UReplayManager_TA* Manager, class FString Id, class UError*
Error);
class AReplayDirector_TA* GetReplayDirector();
bool CanSaveReplay(class UReplay_TA* Replay);
void SaveReplay(class FString SaveName);

```

```

bool ShouldAutoSaveReplay();
};

// Class TAGame.__GFxData_Settings_TA_CrossplayPartyValidation_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Settings_TA_CrossplayPartyValidation_0x1 : public UObject
{
public:
class UGFxData_UserSetting_TA*           UserSetting;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Settings_TA_CrossplayPartyValidation_0x1");
}

return uClassPointer;
};

void __GFxData_Settings_TA_CrossplayPartyValidation_0x4(class UGFxModal_X* _);
void __GFxData_Settings_TA_CrossplayPartyValidation_0x3(class UGFxModal_X* _);
void __GFxData_Settings_TA_CrossplayPartyValidation_0x2(class UGFxModal_X* _);
void __GFxData_Settings_TA_CrossplayPartyValidation_0x1(class UGFxModal_X* _);
};

// Class TAGame.GFxData_UserSetting_TA
// 0x014C (0x0094 - 0x01E0)
class UGFxData_UserSetting_TA : public UGFxDataRow_X
{
public:
uint8_t             Type;          // 0x0098 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FName         Group;        // 0x009C (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FName         Id;           // 0x00A4 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString        Label;        // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString        Description;   // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString        Values;       // 0x00D0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<class FString>    ValuesArray; // 0x00E0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString        Labels;       // 0x00F0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString        Value;        // 0x0100 (0x0010)
[0x0000004040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

```

```

unsigned long          bUserSetting : 1;           // 0x0110 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bEnabled : 1;             // 0x0110 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
class FString          DefaultValue;            // 0x0118 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FProfileSliderLimits    Limits;           // 0x0128 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate     __OnListValueGet__Delegate; // 0x0138
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __OnListValueSet__Delegate; // 0x0150
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __OnBooleanValueGet__Delegate; // 0x0168
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __OnBooleanValueSet__Delegate; // 0x0180
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __OnScalarValueGet__Delegate; // 0x0198
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __OnScalarValueSet__Delegate; // 0x01B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __GetDefaultValue__Delegate;   // 0x01C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_UserSetting_TA");
}

return uClassPointer;
};

void Clear();
int32_t GetClosestListValueInt(int32_t InValue);
void SetScalarValue(class FString InValue);
void SetBooleanValue(class FString InValue, unsigned long bVisualOnly);
void SetListValue(class FString InValue);
void SetValue(class FString InValue);
class UGFxData_UserSetting_TA* AddEnumValue(int32_t InValue, class FString InLabel);
class UGFxData_UserSetting_TA* AddLocalizedValue(class FString InValue, class FString InLabel);
class UGFxData_UserSetting_TA* AddValue(class FString InValue, class FString Key);
void SetScalarLimits(struct FProfileSliderLimits InLimits);
void Init(uint8_t InType, struct FName InGroup, struct FName InID, class FString InDefaultValue,
unsigned long bInUserSetting, unsigned long bInEnabled);
class FString GetDefaultValue();
void OnScalarValueSet(class UGFxData_UserSetting_TA* Setting, float InValue);
float OnScalarValueGet(class UGFxData_UserSetting_TA* Setting);
void OnBooleanValueSet(class UGFxData_UserSetting_TA* Setting, unsigned long InValue);
bool OnBooleanValueGet(class UGFxData_UserSetting_TA* Setting);

```

```

void OnListValueSet(class UGFxData_UserSetting_TA* Setting, class FString InValue);
class FString OnListValueGet(class UGFxData_UserSetting_TA* Setting);
};

// Class TAGame.GFxData_Settings_TA
// 0x0250 (0x0098 - 0x02E8)
class UGFxData_Settings_TA : public UGFxDataSingleton_X
{
public:
    class UTAsyncResult__PrivilegeCheck_X*      CrossNetworkTask;           // 0x0098
    (0x0008) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      MouseSensitivityLimits;          // 0x00A0 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      ControllerDeadzoneLimits;        // 0x00AC
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      MouseDeadZoneLimits;            // 0x00B8
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      KeyboardAxisBlendLimits;       // 0x00C4
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      SteeringSensitivityLimits;     // 0x00D0 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      AirControlSensitivityLimits;   // 0x00DC (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      DodgeInputThresholdLimits;    // 0x00E8
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      SafeZoneLimits;                // 0x00F4 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      UIScaleLimits;                 // 0x0100 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      NameplateScaleLimits;         // 0x010C (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      HDRBrightnessScaleLimits;    // 0x0118
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      HDRPaperWhiteScaleLimits;    // 0x0124
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      HDRGammaScaleLimits;          // 0x0130
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      ForceFeedbackScaleLimits;    // 0x013C
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      TrainingGameSpeedLimits;     // 0x0148
    (0x000C) [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits      FreeplayBoostFillDelayLimits; // 0x0154
    (0x000C) [0x0000000000000001] (CPF_Edit)
    class FString      LoadingString;                  // 0x0160 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class USettingsMenuConfig_TA*      SettingsConfig;             // 0x0170
    (0x0008) [0x0000800000000000]
    class UFreeplayConfig_TA*      FreeplayConfig;               // 0x0178 (0x0008)
    [0x0000800000000000]
    class UItemShopNotificationsConfig_TA*      ItemShopNotificationsConfig; // 0x0180 (0x0008) [0x0000800000000000]
    class FString      ShowBindingsModal_ErrorMessage;          // 0x0188
    (0x0010) [0x000100000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class UGFxData_UserSetting_TA*      CrossplaySetting;          // 0x0198

```

(0x0008) [0x0000000000000000] UserSettings; // 0x01A0  
TArray<class UGFxData\_UserSetting\_TA\*> UserSettings;  
(0x0010) [0x0000000000402000] (CPF\_Transient | CPF\_NeedCtorLink)  
class FString OldResolution; // 0x01B0 (0x0010)  
[0x0000000000402000] (CPF\_Transient | CPF\_NeedCtorLink)  
class FString OldWindowMode; // 0x01C0 (0x0010)  
[0x0000000000402000] (CPF\_Transient | CPF\_NeedCtorLink)  
unsigned long OldVSync : 1; // 0x01D0 (0x0004)  
[0x0000000000002000] [0x00000001] (CPF\_Transient)  
unsigned long bSettingRenderDetail : 1; // 0x01D0 (0x0004)  
[0x0000000000002000] [0x00000002] (CPF\_Transient)  
unsigned long bSettingCameraPreset : 1; // 0x01D0 (0x0004)  
[0x0000000000002000] [0x00000004] (CPF\_Transient)  
class UProfile\_TA\* Profile; // 0x01D8 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class USaveData\_TA\* SaveData; // 0x01E0 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class UGameViewportClient\_X\* Viewport; // 0x01E8 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class UUserSettingObserver\_TA\* SettingObserver; // 0x01F0  
(0x0008) [0x0001000004080009] (CPF\_Edit | CPF\_ExportObject | CPF\_Component |  
CPF\_EditInline)  
class USoundSettingsSave\_TA\* SoundSave; // 0x01F8 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class USoundSettingsControllerSave\_TA\* SoundSaveController; // 0x0200  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UGameplaySettingsSave\_TA\* GameplaySave; // 0x0208  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UProfileGameplaySave\_TA\* ProfileGameplaySave; // 0x0210  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UNetworkSave\_TA\* NetworkSave; // 0x0218 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class UConnectionQualitySave\_TA\* ConnectionQualitySave; // 0x0220  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UProfileCameraSave\_TA\* ProfileCameraSave; // 0x0228  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UProfileGamepadSave\_TA\* GamepadSave; // 0x0230  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UVideoSettingsSave\_TA\* VideoSave; // 0x0238 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class UMusicPlayerSave\_TA\* MusicSave; // 0x0240 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class UProfileControlsSave\_TA\* ProfileControlsSave; // 0x0248  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UNotificationSave\_TA\* NotificationSave; // 0x0250 (0x0008)  
[0x0000000000002000] (CPF\_Transient)  
class UEpicAccountSave\_TA\* EpicAccountSave; // 0x0258  
(0x0008) [0x0000000000002000] (CPF\_Transient)  
class UEOSVoiceSettingsSave\_TA\* VoiceSettingsSave; // 0x0260  
(0x0008) [0x0001000000002000] (CPF\_Transient)  
class UGFxData\_UserSetting\_TA\* SteamInputSettings; // 0x0268  
(0x0008) [0x0001000000002000] (CPF\_Transient)  
int32\_t ControllerCount; // 0x0270 (0x0004)  
[0x0001000000002000] (CPF\_Transient)

```

class FString                               DefaultDeviceString;           // 0x0278 (0x0010)
[0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString                               PermissionLevelNotHighEnoughError; // 0x0288
(0x0010) [0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               TradePermissionNotHighEnoughError; // 0x0298
(0x0010) [0x00000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               WarnCrossPlatformTournament;    // 0x02A8 (0x0010)
[0x00000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               WarnCrossPlatformTournamentAndParty; // 0x02B8
(0x0010) [0x00000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               DisableAndLeave;              // 0x02C8 (0x0010)
[0x00000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               WarnChangingScheduleRegion;   // 0x02D8 (0x0010)
[0x00000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Settings_TA");
}

return uClassPointer;
};

void __GFxData_Settings_TA__InitCameraSettings_0x6(class UGFxData_UserSetting_TA* Setting,
float Value);
float __GFxData_Settings_TA__InitCameraSettings_0x5(class UGFxData_UserSetting_TA* _);
void __GFxData_Settings_TA__InitCameraSettings_0x4(class UGFxData_UserSetting_TA* Setting,
float Value);
float __GFxData_Settings_TA__InitCameraSettings_0x3(class UGFxData_UserSetting_TA* _);
void __GFxData_Settings_TA__InitCameraSettings_0x2(class UGFxData_UserSetting_TA* Setting,
unsigned long Value);
bool __GFxData_Settings_TA__InitCameraSettings_0x1(class UGFxData_UserSetting_TA* _);
void __GFxData_Settings_TA__InitNetworkSettings_0x2(class UGFxData_UserSetting_TA*
Setting, class FString Value);
class FString __GFxData_Settings_TA__InitNetworkSettings_0x1(class
UGFxData_UserSetting_TA* Setting);
void __GFxData_Settings_TA__InitGameplaySettings_0x4(class UGFxData_UserSetting_TA* _,
class FString InValue);
class FString __GFxData_Settings_TA__InitGameplaySettings_0x3(class
UGFxData_UserSetting_TA* _);
void __GFxData_Settings_TA__InitGameplaySettings_0x2(class UGFxData_UserSetting_TA* _,
unsigned long bValue);
bool __GFxData_Settings_TA__InitGameplaySettings_0x1(class UGFxData_UserSetting_TA* _);
void __GFxData_Settings_TA__SetTourScheduleRegion_0x2(class UGFxModal_X* _);
void __GFxData_Settings_TA__SetDefaults_0x2(class UGFxData_UserSetting_TA* Setting);
void TriggerSettingChangedEvent(struct FName SettingName);
void InitReplayFXSettings();
void UpdateBoostFillDelaySettingEnabled(class UGFxData_UserSetting_TA* UserSetting);
void SetFreeplayBoostFillDelay(class UGFxData_UserSetting_TA* UserSetting, float Value);

```

```
float GetFreeplayBoostFillDelay(class UGFxData_UserSetting_TA* UserSetting);
void SetDisableFreeplayGoals(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetDisableFreeplayGoals(class UGFxData_UserSetting_TA* UserSetting);
void SetTrainingGameSpeed(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetTrainingGameSpeed(class UGFxData_UserSetting_TA* UserSetting);
void SetDodgeInputThreshold(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetDodgeInputThreshold(class UGFxData_UserSetting_TA* UserSetting);
void SetKeyboardAxisBlendTime(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetKeyboardAxisBlendTime(class UGFxData_UserSetting_TA* UserSetting);
void SetControllerDeadzone(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetControllerDeadzone(class UGFxData_UserSetting_TA* UserSetting);
void SetMouseSensitivity(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetMouseSensitivity(class UGFxData_UserSetting_TA* UserSetting);
void SetAirControlSensitivity(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAirControlSensitivity(class UGFxData_UserSetting_TA* UserSetting);
void SetSteeringSensitivity(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetSteeringSensitivity(class UGFxData_UserSetting_TA* UserSetting);
void UpdateForceFeedbackEnabled(class UGFxData_UserSetting_TA* UserSetting);
void SetForceFeedbackMode(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetForceFeedbackMode(class UGFxData_UserSetting_TA* UserSetting);
void SetForceFeedbackScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetForceFeedbackScale(class UGFxData_UserSetting_TA* UserSetting);
void ShowBindings();
void OnControllerRegistrationChanged(int32_t LocalPlayerNum);
void HandleUnregisteredController(int32_t LocalPlayerNum);
void HandleRegisteredController(int32_t LocalPlayerNum, int32_t ControllerId);
void RefreshSteamInputSettings();
class FString SteamInputSettings_ControllerSelected(int32_t ControllerId);
class FString SteamInputSettings_IgnoreInput();
class FString SteamInputSettings_Autodetect();
int32_t GetControllerIDFromValue(class FString Value);
void SetSteamInputSettings(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetSteamInputSettings(class UGFxData_UserSetting_TA* UserSetting);
void SetAirPitchSafetyEnabled(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetAirPitchSafetyEnabled(class UGFxData_UserSetting_TA* UserSetting);
void SetControlPreset(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetControlPreset(class UGFxData_UserSetting_TA* UserSetting);
void AddToControlPresets(struct FName PresetName, TArray<struct FBindingPreset>& Presets);
TArray<struct FBindingPreset> GetControlPresets();
void OnControlPresetChanged();
void SetOutputAudioDevice(class UGFxData_UserSetting_TA* UserSetting, class FString DeviceID);
class FString GetOutputAudioDevice(class UGFxData_UserSetting_TA* UserSetting);
void SetInputAudioDevice(class UGFxData_UserSetting_TA* UserSetting, class FString DeviceID);
class FString GetInputAudioDevice(class UGFxData_UserSetting_TA* UserSetting);
void SetOutputVoiceVolume(class UGFxData_UserSetting_TA* UserSetting, float OutputVolume);
float GetOutputVoiceVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetPreferredVoiceRoom(class UGFxData_UserSetting_TA* UserSetting, class FString PreferredType);
class FString GetPreferredVoiceRoom(class UGFxData_UserSetting_TA* UserSetting);
void SetVoiceTextNotifications(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
```

```
bool GetVoiceTextNotifications(class UGFxData_UserSetting_TA* UserSetting);
void SetVoiceMatchNotifications(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetVoiceMatchNotifications(class UGFxData_UserSetting_TA* UserSetting);
void SetVoiceMenuNotifications(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetVoiceMenuNotifications(class UGFxData_UserSetting_TA* UserSetting);
void SetVoiceInputMode(class UGFxData_UserSetting_TA* UserSetting, class FString VoiceInputMode);
class FString GetVoiceInputMode(class UGFxData_UserSetting_TA* UserSetting);
void InitControlsSettings();
bool IsKnockoutMode();
bool IsAutoCamSpectator();
bool IsSpectator();
void SetDefaults(struct FName Group);
int32_t CompareSettingDefaultPriority(class UGFxData_UserSetting_TA* A, class UGFxData_UserSetting_TA* B);
class FString GetCrossplayDefaultValue();
void SetCrossPlay(class UGFxData_UserSetting_TA* UserSetting, unsigned long bNewValue);
void CrossplayPartyValidation(class UGFxData_UserSetting_TA* UserSetting, unsigned long RequestedValue);
void HandleCrossPlatformEnabledChanged();
bool GetCrossPlay(class UGFxData_UserSetting_TA* UserSetting);
void SetNameplateScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetNameplateScale(class UGFxData_UserSetting_TA* UserSetting);
void SetNameplateMode(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetNameplateMode(class UGFxData_UserSetting_TA* UserSetting);
void SetShowItemShopNotifications(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetShowItemShopNotifications(class UGFxData_UserSetting_TA* UserSetting);
void SetShowInGameNotifications(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetShowInGameNotifications(class UGFxData_UserSetting_TA* UserSetting);
void SetCompDivisions(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetCompDivisions(class UGFxData_UserSetting_TA* UserSetting);
void SetMessageLevel(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetMessageLevel(class UGFxData_UserSetting_TA* UserSetting);
void SetMetric(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetMetric(class UGFxData_UserSetting_TA* UserSetting);
void SetBallIndicator(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetBallIndicator(class UGFxData_UserSetting_TA* UserSetting);
void UpdateEnableSavingValue();
void HandleSaveDataPostEnableComplete(class USaveGameManager_TA* SaveManager, class USaveData_TA* ForSaveData, class UError* Error);
void RefreshTourScheduleRegionLabel();
void HandleFailedSetTourScheduleRegion(class UError* InError);
void SetTourScheduleRegion(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetTourScheduleRegion(class UGFxData_UserSetting_TA* UserSetting);
void SetInGameStatNotification(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetInGameStatNotification(class UGFxData_UserSetting_TA* UserSetting);
void SetEnableSaving(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetEnableSaving(class UGFxData_UserSetting_TA* UserSetting);
void SetSplitscreenMode(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
```

```
class FString GetSplitscreenMode(class UGFxData_UserSetting_TA* UserSetting);
void SetTrainingControlsVisibility(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetTrainingControlsVisibility(class UGFxData_UserSetting_TA* UserSetting);
void SetFreeplayBoostFillType(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetFreeplayBoostFillType(class UGFxData_UserSetting_TA* UserSetting);
void SetPartyChatFilter(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetPartyChatFilter(class UGFxData_UserSetting_TA* UserSetting);
void SetMatchChatFilter(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetMatchChatFilter(class UGFxData_UserSetting_TA* UserSetting);
void SetQuickChatFilter(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetQuickChatFilter(class UGFxData_UserSetting_TA* UserSetting);
void SetTacticalChatFilter(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetTacticalChatFilter(class UGFxData_UserSetting_TA* UserSetting);
void UpdateTacticalChatSettingEnabled(class UGFxData_UserSetting_TA* UserSetting);
void SetVoiceChatFilter(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetVoiceChatFilter(class UGFxData_UserSetting_TA* UserSetting);
void SetTeamColoredBoostMeter(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetTeamColoredBoostMeter(class UGFxData_UserSetting_TA* UserSetting);
void SetStatGraphLevel(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetStatGraphLevel(class UGFxData_UserSetting_TA* UserSetting);
void SetConvertPlatformFriends(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetConvertPlatformFriends(class UGFxData_UserSetting_TA* UserSetting);
void InitInterfaceSettings();
void InitChatSettings();
void SetChatPermissionDefault(uint8_t InLevel, class UGFxData_UserSetting_TA*& InSetting);
void InitTrainingSettings();
void InitGameplaySettings();
void SetShowNetQualityIndicators(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetShowNetQualityIndicators(class UGFxData_UserSetting_TA* Setting);
void SetIncomingBandwidth(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetIncomingBandwidth(class UGFxData_UserSetting_TA* Setting);
void SetIncomingPacketFrequency(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetIncomingPacketFrequency(class UGFxData_UserSetting_TA* Setting);
void SetOutgoingPacketFrequency(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetOutgoingPacketFrequency(class UGFxData_UserSetting_TA* Setting);
void InitNetworkSettings();
void InitCrossPlaySetting();
void SetMaxProximityDistance(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetMaxProximityDistance(class UGFxData_UserSetting_TA* UserSetting);
bool GetFocusOnGoals(class UGFxData_UserSetting_TA* UserSetting);
void SetFocusOnGoals(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetUseReplayCamera(class UGFxData_UserSetting_TA* UserSetting);
void SetUseReplayCamera(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
float GetAutoCamAerialPrediction(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamAerialPrediction(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamPlayerFocusBlendSpeed(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamPlayerFocusBlendSpeed(class UGFxData_UserSetting_TA* UserSetting, float
```

```
Value);
float GetAutoCamAerialDistance(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamAerialDistance(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamRotateSpeed(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamRotateSpeed(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamMoveSpeedZ(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamMoveSpeedZ(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamMoveSpeed(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamMoveSpeed(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamZ(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamZ(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamDistance(class UGFxData_UserSetting_TA* UserSetting);
void SetAutoCamDistance(class UGFxData_UserSetting_TA* UserSetting, float Value);
void SetAutoCamFOV(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAutoCamFOV(class UGFxData_UserSetting_TA* UserSetting);
void SetBallcamMode(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetBallcamMode(class UGFxData_UserSetting_TA* UserSetting);
void SetBallcamIndicator(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetBallcamIndicator(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraType(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetCameraType(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraTransitionSpeed(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraTransitionSpeed(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraSwivelSpeed(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraSwivelSpeed(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraStiffness(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraStiffness(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraDistance(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraDistance(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraAngle(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraAngle(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraHeight(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraHeight(class UGFxData_UserSetting_TA* UserSetting);
void SetGreenScreenColor(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetGreenScreenColor(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraBloomScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraBloomScale(class UGFxData_UserSetting_TA* UserSetting);
class FString GetImageOverlaySetting(class UGFxData_UserSetting_TA* UserSetting);
class FString GetImageFilterSetting(class UGFxData_UserSetting_TA* UserSetting);
void SetFilterOverlayIntensity(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetFilterOverlayIntensity(class UGFxData_UserSetting_TA* UserSetting);
void SetImageFilterIntensity(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetImageFilterIntensity(class UGFxData_UserSetting_TA* UserSetting);
void SetImageOverlaySetting(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
void SetImageFilterSetting(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
void SetReplayTranslucent(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetReplayTranslucent(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraDOFKernel(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraDOFKernel(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraDOFMaxNear(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraDOFMaxNear(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraDOFMaxFar(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraDOFMaxFar(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraDOFDistance(class UGFxData_UserSetting_TA* UserSetting, float Value);
```

```
float GetCameraDOF(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraFOV(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCameraFOV(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraShake(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetCameraShake(class UGFxData_UserSetting_TA* UserSetting);
void SetLockSpectatorBallCam(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetLockSpectatorBallCam(class UGFxData_UserSetting_TA* UserSetting);
void SetInvertSpectatorPitch(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetInvertSpectatorPitch(class UGFxData_UserSetting_TA* UserSetting);
void SetInvertSwivelPitch(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetInvertSwivelPitch(class UGFxData_UserSetting_TA* UserSetting);
void SetCameraSettingsPreset(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
void SaveCameraSettingsPreset(class UGFxData_UserSetting_TA* UserSetting);
class FString GetCameraSettingsPreset(class UGFxData_UserSetting_TA* UserSetting);
void OnCameraOptionChanged();
void InitCameraSettings();
void SetPushToTalk(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetPushToTalk(class UGFxData_UserSetting_TA* UserSetting);
void SetStreamerSafeAudioEnabled(class UGFxData_UserSetting_TA* UserSetting, unsigned long bnewValue);
bool GetStreamerSafeAudioEnabled(class UGFxData_UserSetting_TA* UserSetting);
void SetMuteOnFocusLost(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetMuteOnFocusLost(class UGFxData_UserSetting_TA* UserSetting);
void SetMusicStingersSetting(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetMusicStingersSetting(class UGFxData_UserSetting_TA* UserSetting);
void SetPlayMusicInGameSetting(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetPlayMusicInGameSetting(class UGFxData_UserSetting_TA* UserSetting);
void SetPlayMusicInTrainingSetting(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetPlayMusicInTrainingSetting(class UGFxData_UserSetting_TA* UserSetting);
void SetPlayMusicInMenuSetting(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetPlayMusicInMenuSetting(class UGFxData_UserSetting_TA* UserSetting);
void SetDynamicRange(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetDynamicRange(class UGFxData_UserSetting_TA* UserSetting);
void SetOutputType(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetOutputType(class UGFxData_UserSetting_TA* UserSetting);
void SetControllerSoundMode(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetControllerSoundMode(class UGFxData_UserSetting_TA* UserSetting);
void SetControllerVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetControllerVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetVoiceVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetVoiceVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetCrowdVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetCrowdVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetAmbientVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetAmbientVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetGameplayMusicVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
```

```
float GetGameplayMusicVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetMusicVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetMusicVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetGameVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetGameVolume(class UGFxData_UserSetting_TA* UserSetting);
void SetMasterVolume(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetMasterVolume(class UGFxData_UserSetting_TA* UserSetting);
void InitAudioSettings();
void SetUncappedFramerate(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetUncappedFramerate(class UGFxData_UserSetting_TA* UserSetting);
void SetUIScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetUIScale(class UGFxData_UserSetting_TA* UserSetting);
void SetSafeZone(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetSafeZone(class UGFxData_UserSetting_TA* UserSetting);
void SetMotionBlur(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetMotionBlur(class UGFxData_UserSetting_TA* UserSetting);
void SetDynamicShadows(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetDynamicShadows(class UGFxData_UserSetting_TA* UserSetting);
void SetLensFlares(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetLensFlares(class UGFxData_UserSetting_TA* UserSetting);
void SetLightShaftsAdvancedOption(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
void SetLightShafts(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetLightShafts(class UGFxData_UserSetting_TA* UserSetting);
void SetBloom(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetBloom(class UGFxData_UserSetting_TA* UserSetting);
void SetDepthOfField(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetDepthOfField(class UGFxData_UserSetting_TA* UserSetting);
void SetAmbientOcclusion(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetAmbientOcclusion(class UGFxData_UserSetting_TA* UserSetting);
void SetHighQualityShaders(class UGFxData_UserSetting_TA* UserSetting, unsigned long Value);
bool GetHighQualityShaders(class UGFxData_UserSetting_TA* UserSetting);
void SetWorldDetail(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetParticleDetail(class UGFxData_UserSetting_TA* UserSetting);
class FString GetWorldDetail(class UGFxData_UserSetting_TA* UserSetting);
void SetParticleDetail(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
void SetTextureDetail(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
void SetRenderDetailValue(class FString Value);
void SetRenderDetail(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
void SaveVideoValueString(class UGFxData_UserSetting_TA* UserSetting);
class FString GetSavedVideoValueString(class UGFxData_UserSetting_TA* UserSetting);
void SetRenderQuality(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetRenderQuality(class UGFxData_UserSetting_TA* UserSetting);
void SetAntiAlias(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetAntiAlias(class UGFxData_UserSetting_TA* UserSetting);
void OnAdvancedVideoOptionChanged();
void InitVideoSettings();
void SetEnableHDRSideBySideVisualizer(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetEnableHDRSideBySideVisualizer(class UGFxData_UserSetting_TA* UserSetting);
void SetHDRGammaScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetHDRGammaScale(class UGFxData_UserSetting_TA* UserSetting);
void SetHDRPaperWhiteScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
```

```
float GetHDRPaperWhiteScale(class UGFxData_UserSetting_TA* UserSetting);
void SetHDRBrightnessScale(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetHDRBrightnessScale(class UGFxData_UserSetting_TA* UserSetting);
void SaveViewportSettings();
void RevertViewportSettings();
void SetMSAASampleCountScale(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetMSAASampleCountScale(class UGFxData_UserSetting_TA* UserSetting);
void SetViewportSettings(class FString ResolutionValue, class FString WindowModeValue,
unsigned long VSyncValue);
void ApplyViewportSettings(class FString ResolutionValue, class FString WindowModeValue,
unsigned long VSyncValue);
void SetRefreshRate(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetRefreshRate(class UGFxData_UserSetting_TA* UserSetting);
void SetMaxFPS(class UGFxData_UserSetting_TA* UserSetting, float Value);
float GetMaxFPS(class UGFxData_UserSetting_TA* UserSetting);
void SetForceDefaultColors(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetForceDefaultColors(class UGFxData_UserSetting_TA* UserSetting);
void SetEffectIntensity(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetEffectIntensity(class UGFxData_UserSetting_TA* UserSetting);
void SetColorBlind(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetColorBlind(class UGFxData_UserSetting_TA* UserSetting);
void SetConsoleQualityMode(class UGFxData_UserSetting_TA* UserSetting, class FString Value);
class FString GetConsoleQualityMode(class UGFxData_UserSetting_TA* UserSetting);
void SetWeatherFX(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetWeatherFX(class UGFxData_UserSetting_TA* UserSetting);
void SetVSync(class UGFxData_UserSetting_TA* UserSetting, unsigned long bValue);
bool GetVSync(class UGFxData_UserSetting_TA* UserSetting);
class FString GetWindowMode(class UGFxData_UserSetting_TA* UserSetting);
class FString GetResolution(class UGFxData_UserSetting_TA* UserSetting);
void InitViewportSettings();
int32_t SortResolutions(struct FIntPoint A, struct FIntPoint B);
int32_t GetGreatestCommonDenominator(int32_t X, int32_t Y);
void SetScalarValue(struct FName Id, float Value);
void SetBooleanValue(struct FName Id, unsigned long Value);
void SetListValue(struct FName Id, class FString Value);
class FString EnumToString(int32_t Value);
class FString BoolToString(unsigned long Value);
bool StringToBool(class FString Value);
class UGFxData_UserSetting_TA* CreateScalarUserSetting(struct FName Group, struct FName Id,
float DefaultValue, struct FScriptDelegate GetFunc, struct FScriptDelegate SetFunc, struct
FProfileSliderLimits Limits, unsigned long bUserSetting, unsigned long bEnabled);
class UGFxData_UserSetting_TA* CreateBooleanUserSetting(struct FName Group, struct FName Id,
unsigned long DefaultValue, struct FScriptDelegate GetFunc, struct FScriptDelegate SetFunc,
unsigned long bUserSetting, unsigned long bEnabled);
class UGFxData_UserSetting_TA* CreateListUserSetting(struct FName Group, struct FName Id,
class FString DefaultValue, struct FScriptDelegate GetFunc, struct FScriptDelegate SetFunc,
unsigned long bUserSetting, unsigned long bEnabled);
class UGFxData_UserSetting_TA* GetUserSetting(struct FName Id);
int32_t GetUserSettingIndex(struct FName Id);
void LoadUserSettings();
void eventOnRemoved();
void eventOnShellSet();
```

```

};

// Class TAGame.__GFxData_Settings_TA_SetDefaults_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_Settings_TA_SetDefaults_0x1 : public UObject
{
public:
struct FName           Group;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Settings_TA_SetDefaults_0x1");

}

return uClassPointer;
};

bool __GFxData_Settings_TA_SetDefaults_0x1(class UGFxData_UserSetting_TA* Setting);

// Class TAGame.__GFxData_Settings_TA_SetTourScheduleRegion_0x1
// 0x0018 (0x0060 - 0x0078)
class U_GFxData_Settings_TA_SetTourScheduleRegion_0x1 : public UObject
{
public:
class UAutoTour_TA*      AutoTour;     // 0x0060 (0x0008)
[0x0001000000000000]
class FString            Value;        // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Settings_TA_SetTourScheduleRegion_0x1");

}

return uClassPointer;
};

void __GFxData_Settings_TA_SetTourScheduleRegion_0x1(class UGFxModal_X* _);

// Class TAGame.__GFxData_ShopCatalogue_TA_GetOnlineProductsByCostID_0x1

```

```

// 0x0004 (0x0060 - 0x0064)
class U_GFxDATA_ShopCatalogue_TA__GetOnlineProductsByCostID_0x1 : public UObject
{
public:
int32_t CostID; // 0x0060 (0x0004)
[0x0010000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_ShopCatalogue_TA__GetOnlineProductsByCostID_0x1");
}

return uClassPointer;
};

bool __GFxDATA_ShopCatalogue_TA__GetOnlineProductsByCostID_0x1(struct FShopItem I);
};

// Class TAGame._GFxDATA_ShopCatalogue_TA__GetShopItemByCostID_0x1
// 0x00EC04 (0x009460 - 0x0180064)
class U_GFxDATA_ShopCatalogue_TA : public UGFxDATARow_X
{
public:
class UShopsManager_TA* ShopManager; // 0x0098 (0x0008)
[0x00180000000001] (CPF_Edit) __GetShopItemByCostID_0x1 : public UObject
{
public:
int32_t ShopCostID; // 0x00A60 (0x0004)
[0x0010000400020000] (CPF_Transient | CPF_EditInlineNotify)
class FString ShopName; // 0x00A8 (0x0010)
[0x001000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString Type; // 0x00B8 (0x0010)
[0x001000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString Region; // 0x00C8 (0x0010)
[0x001000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
uint64_t StartDate; // 0x00D8 (0x0008)
[0x001000040002000] (CPF_Transient | CPF_EditInlineNotify)
uint64_t EndDate; // 0x00E0 (0x0008)
[0x001000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture* ShopImage; // 0x00E8 (0x0008)
[0x001000040002000] (CPF_Transient | CPF_EditInlineNotify)
class FString ShopImageURL; // 0x00F0 (0x0010)
[0x001000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long bLoaded : 1; // 0x0100 (0x0004)
[0x0009000040002000] [0x00000001] (CPF_Transient | CPF_EditInlineNotify)
unsigned long bFailedToLoad : 1; // 0x0100 (0x0004)
[0x0001000040002000] [0x00000002] (CPF_Transient | CPF_EditInlineNotify)
unsigned long bShopItemExpired : 1; // 0x0100 (0x0004)

```

```

[0x0001000040002000] [0x00000004] (CPF_Transient | CPF_EditInlineNotify)
unsigned long          bAutoReload : 1;           // 0x0100 (0x0004)
[0x0009000040002000] [0x00000008] (CPF_Transient | CPF_EditInlineNotify)
unsigned long          bNewCatalogue : 1;        // 0x0100 (0x0004)
[0x0001000040002000] [0x00000010] (CPF_Transient | CPF_EditInlineNotify)
TArray<struct FShopItem>      CachedItems;          // 0x0108 (0x0010)
[0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
float                  CacheDuration;          // 0x0118 (0x0004)
[0x0001000000004000] (CPF_Config)
float                  LastCacheTime;          // 0x011C (0x0004)
[0x0001000000002000] (CPF_Transient)
float                  LastOwnedCacheTime;    // 0x0120 (0x0004)
[0x0001000000002000] (CPF_Transient00)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_ShopCatalogue_TA__GetShopItemByCostID_0x1");
}

return uClassPointer;
};

bool __GFxDATA_ShopCatalogue_TA__GetShopItemByCostID_0x1(struct FShopItem I);

// Class TAGame._GFxDATA_ShopCatalogue_TA__LoadCatalogueInternal_0x1
// 0x0004 (0x0060 - 0x0064)
class UAsyncTask*          ShopTask;            // 0x0128 (0x0008)
[0x0001000000002000] (CPF_Transient)
class UOnlineProductStoreSet_TA* CatalogueSet; // 0x0130
(0x0008) [0x000100004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UShopItemExpirationHelper_TA* ExpirationHelper; // 0x0138
(0x0008) [0x000100000002000] (CPF_Transient)
class UShopAutoLoadHelper_TA* AutoLoadHelper; // 0x0140
(0x0008) [0x000100000002000] (CPF_Transient)
class UShopsConfig_TA*     ShopsConfig;

_GFxDATA_ShopCatalogue_TA__LoadCatalogueInternal_0x1 : public UObject
{
public:
float                  RequestStartTime;       // 0x0148060 (0x00084)
[0x0001800000000000]
struct FScriptDelegate      _bLoaded_ChangeNotify; // 0x0150
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      _bAutoReload_ChangeNotify; // 0x0168
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ShopCatalogue_TA");
}

return uClassPointer;
};

void __GFxData_ShopCatalogue_TA__OnShellSet_0x1(class UShopItemExpirationHelper_TA* _);
int32_t __GFxData_ShopCatalogue_TA__ContainsOnlineProducts_0x2(class UOnlineProduct_TA* P);
int32_t __GFxData_ShopCatalogue_TA__ContainsOnlineProducts_0x1(struct FOnlineProductData P);
struct FProductHashID __GFxData_ShopCatalogue_TA__GetOnlineProductsByCostID_0x2(struct FShopDeliverable D);
bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x3(struct FShopDeliverable D);
bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x2(struct FShopDeliverable P);
bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x1(class UOnlineProduct_TA* P);
void __bAutoReload__ChangeNotifyFunc();
void __bLoaded__ChangeNotifyFunc();
bool IsActive(uint64_t CurrentTime);
void DebugSetCataloguelItems(TArray<struct FShopItem> InItems);
void DebugOverride(int32_t Index, int32_t ProductID, int32_t PaintID, int32_t CertifiedId, int32_t SpecialEditionID, int32_t Cost);
void OnCatalogueLoaded();
void HandleShopItemExpired();
void HandleCatalogueLoadFailed(class UError* Error);
void ShopItemTextureLoaded(int32_t ShopItemID, class UTexture* LoadedTexture);
void SyncItemImages(class UWebImageCache_X* WebImageCache, int32_t I);
void HandleCatalogueLoaded(TArray<struct FShopItem> InItems);
void SetIsNewCatalogue(unsigned long blnNewCatalogue);
void SetOwnedDeliverableProducts();
bool IsDeliverableSubset(TArray<struct FShopDeliverable>& Superset, TArray<struct FShopDeliverable>& Subset);
void ClearLastOwnedCacheTime();
void SetCataloguelItems(TArray<struct FShopItem> InItems);
TArray<class UOnlineProduct_TA*> GetOnlineProductsByCostID(int32_t CostID);
bool ContainsOnlineProducts(TArray<struct FOnlineProductData>& Products);
void LoadCatalogueInternal(unsigned long bForce);
void SetAutoLoadCatalogue(unsigned long blnAutoReload);
00]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ShopCatalogue_TA__LoadCatalogueInternal_0x1");
}

return uClassPointer;
};

void __GFxData_ShopCatalogue_TA__LoadCatalogueInternal_0x1(TArray<struct FShopItem>
GetShopItems();
void ForceLoadCatalogue();
void LoadCatalogue();
void Init(struct FShopData InData);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.__GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x4
// 0x0018 (0x0060 - 0x0078)
class U__GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x4 : public UObject
{
public:
int32_t ShopItemID; // 0x0060 (0x0004)
[0x0001000000000000]
TArray<struct FShopDeliverable> DeliverableProducts; // 0x0068
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x4");
}

return uClassPointer;
};

bool __GFxData_ShopCatalogue_TA__SetOwnedDeliverableProducts_0x4(struct FShopItem I);
};

// Class TAGame.__GFxData_ShopCatalogue_TA__SyncItemImages_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GFxData_ShopCatalogue_TA__SyncItemImages_0x1 : public UObject
{
public:
int32_t ShopItemID; // 0x0060 (0x0004)
[0x0001000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_ShopCatalogue_TA__SyncItemImages_0x1");
}

return uClassPointer;
};

void __GFxData_ShopCatalogue_TA__SyncItemImages_0x1(class UTexture2DDynamic* Texture);
};

// Class TAGame._GFxData_Shops_TA__GetCrateShop_0x1
// 0x0004 (0x0060 - 0x0064)
class U_GFxData_Shops_TA__GetCrateShop_0x1 : public UObject
{
public:
int32_t CrateID; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._GFxData_Shops_TA__GetCrateShop_0x1");
}

return uClassPointer;
};

void __GFxData_Shops_TA__GetCrateShop_0x1(TArray<struct FShopItem> Items, int32_t ShopID);
};

// Class TAGame._GFxData_Shops_TA
// 0x0080 (0x0098 - 0x0118)
class UGFxData_Shops_TA : public UGFxDataSingleton_X
{
public:
class UShopsManager_TA* ShopManager; // 0x0098 (0x0008)
[0x0001800000000001] (CPF_Edit)
class UOnlineImageDownloaderWeb* ImageDownloader;
__HandlePurchaseSuccess_0x1
// 0xA04 (0x0008) [0x0001800000000001] (CPF_Edit) 60 - 0x0064
class UShopsConfig_TA* Config; // 0x00A8 (0x0008)
[0x0001800000002000] (CPF_Transient)
TArray<class UGFxData_ShopCatalogue_TA*> Shops; // 0x00B0

```

```

(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t DebugShopID; // 0x00C0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
unsigned long bHideMainMenuButton : 1; // 0x00C4 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long bNewCatalogues : 1; // 0x00C4 (0x0004)
[0x0001000040000000] [0x00000002] (CPF>EditInlineNotify)
class UOnlineProductStoreSet_TA* CrateShopSet; // 0x00C8
(0x0008) [0x000100004082008] (CPF>ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
float CacheDuration; // 0x00D0 (0x0004)
[0x000100000004000] (CPF>Config)
float LastCacheTime; // 0x00D4 (0x0004)
[0x000100000002000] (CPF_Transient)
class UAsyncTask* ShopTask; // 0x00D8 (0x0008)
[0x000100000002000] (CPF_Transient)
class UShopMetrics_TA* ShopMetrics; // 0x00E0 (0x0008)
[0x0001000004080008] (CPF>ExportObject | CPF_Component | CPF>EditInline)
class UShopTabs_TA* ShopTabs; // 0x00E8 (0x0008)
[0x0001004004080008] (CPF>ExportObject | CPF_Component | CPF>EditInline)
class UShopNotificationsManager_TA*
ShopNotificationsManager;__GFxData_Shops_TA__HandlePurchaseSuccess_0x1 : public
UObject
{
public:
int32_t CostID; // 0x00F60 (0x00084)
[0x0001004000000000]
class UShopCatalogueCacheSave_TA* CatalogueCache; // 0x00F8
(0x0008) [0x0001008000002000] (CPF_Transient)
struct FScriptDelegate __EventReceivedShops__Delegate; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Shops_TA");
}

return uClassPointer;
};

void __GFxData_Shops_TA__OnShellSet_0x1(class UShopCatalogueCacheSave_TA* InCache);
class UGFxData_ShopCatalogue_TA* __GFxData_Shops_TA__BuildShopCatalogues_0x1(struct
FShopData Data);
class FString __GFxData_Shops_TA__HandleGetAllShops_0x2(struct FShopData S);
bool __GFxData_Shops_TA__HandleGetAllShops_0x1(struct FShopData S);
bool __GFxData_Shops_TA__RefreshShopsViewed_0x1(class UGFxData_ShopCatalogue_TA* S);
bool __GFxData_Shops_TA__SyncFeaturedProducts_0x1(class UGFxData_ShopCatalogue_TA*
S);
void __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x5(class UGFxModal_X* _);

```

```
void __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x4(class UGFxModal_X* _);
void __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x3(class UGFxModal_X* _);
bool __GFxData_Shops_TA__OnCheckForPurchaseWarning_0x2(class UOnlineProduct_TA* P);
TArray<struct FShopTab> GetAvailableShopTabs();
static void CreateTemporaryOnlineProducts(class UOnlineProductStoreSet_TA* ProductSet,
TArray<struct FShopItem>& Items);
void OnGetCrateShopFailed();
void HandleGetCrateShopFailed(class UError* Error);
void OnReceivedCrateShop(int32_t CrateID, TArray<struct FShopItem> CrateItems, int32_t ShopID);
void GetCrateShop(int32_t CrateID);
void OnPurchaseFail();
void HandlePurchaseFail(class UError* Error);
void OnPurchaseSuccess();
void HandlePurchaseSuccess(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count,
TArray<struct FOnlineProductData> PurchasedProducts, TArray<struct FCurrency>
PurchasedCurrencies);
void OnPurchase(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count);
void OnCheckForPurchaseWarningSuccess();
bool OnCheckForPurchaseWarning(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count);
void CheckForPurchaseWarning(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count);
bool CheckPlayerMeetsProductRequirements(class UOnlineProduct_TA* OnlineProduct,
TArray<class UOnlineProduct_TA*>& DeliverableOnlineProducts, class FString& ErrorString);
void HandleShopItemTextureLoaded(int32_t ShopID, int32_t ShopItemIndex, class UTexture* LoadedTexture);
void Purchase(int32_t CostID, int32_t ShopID, int32_t InDiscountID, int32_t Count);
void PreviewShopItem(int32_t ShopItemID);
void ExitShopPage();
void OnMTXShopChanged(int32_t NewCatalogID);
void OnShopCatalogueChanged(int32_t NewShopID);
void OnEnterShopMenu(int32_t BrowseSourceID);
void SyncFeaturedProducts(TArray<struct FOnlineProductData>& Products);
void RefreshOwnedProducts();
void RefreshShopsViewed();
void SetShopViewed(int32_t InShopID);
int32_t GetShopCatalogueIndex(int32_t InShopID);
bool CheckShopExists(int32_t InShopID);
void OnGetAllShopsFailed();
void OnGetAllShopsSuccess();
void HandleGetAllShopsFailed(class UError* Error);
void HandleImageDownloaded(struct FOnlineImageDownload DownloadResult);
void HandleGetAllShops(TArray<struct FShopData> InShops);
TArray<class UGFxData_ShopCatalogue_TA*> BuildShopCatalogues(TArray<struct FShopData>&
InShops);
void GetAllShopsInternal(unsigned long bForce);
void GetAllShopsForced();
void RefreshAllShops();
void HandlePsyNetLogin();
void HandleConfigChanged();
void eventOnRemoved();
void eventOnShellSet();
void EventReceivedShops(class UGFxData_Shops_TA* GFxShops00]
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Shops_TA__HandlePurchaseSuccess_0x1");
}

return uClassPointer;
};

bool __GFxData_Shops_TA__HandlePurchaseSuccess_0x1(struct FShopItem I);
};

// Class TAGame._GFxData_Shops_TA__HandlePurchaseSuccessLoadAllShopCatalogues_0x1
// 0x000814 (0x0060 - 0x006874)
class U_GFxData_Shops_TA__HandlePurchaseSuccessLoadAllShopCatalogues_0x1 : public
UObject
{
public:
TArray<int32_t > ShopID; RequestIDs; // 0x0060
(0x00104) [0x00010000000400000]
int32_(CPF_NeedCtorLink)
float CostID; RequestStartTime; // 0x006470
(0x0004) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_Shops_TA__HandlePurchaseSuccessLoadAllShopCatalogues_0x1");
}

return uClassPointer;
};

boolvoid __GFxData_Shops_TA__HandlePurchaseSuccess_0x2(struct FShopItem
ILoadAllShopCatalogues_0x2(class UError* Error);
boolvoid __GFxData_Shops_TA__HandlePurchaseSuccess_0x1(class
UGFxData_LoadAllShopCatalogues_0x1(TArray<struct FShopCatalogue_TA* S> Catalogues);
};

// Class
TAGame._GFxData_Shops_TA__OnCheckForPurchaseWarningLoadCataloguesByIndex_0x1
// 0x00014 (0x0060 - 0x00674)
class U_GFxData_Shops_TA__OnCheckForPurchaseWarningLoadCataloguesByIndex_0x1 :
```

```
public UObject
{
public:
TArray<int32_t>           ShopID;      RequestIDs;          // 0x0060 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
float                      RequestStartTime;    // 0x00670 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDatas_Shops_TA__OnCheckForPurchaseWarningLoadCataloguesByIndex_0x1");

}

return uClassPointer;
};

bool void
__GFxDatas_Shops_TA__OnCheckForPurchaseWarningLoadCataloguesByIndex_0x12(class
UGFxDatas_ShopCatalogue_TA* SError* Error);
void __GFxDatas_Shops_TA__LoadCataloguesByIndex_0x1(TArray<struct FShopCatalogue>
Catalogues);
};

// Class TAGame._GFxDatas_Shops_TA__OnPurchase_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxDatas_Shops_TA__OnPurchase_0x1 : public UObject
{
public:
int32_t           CostID;      // 0x0060 (0x0004)
[0x0001000000000000]
int32_t           ShopID;      // 0x0064 (0x0004)
[0x0001000000000000]
int32_t           InDiscountID; // 0x0068 (0x0004)
[0x0001000000000000]
int32_t           Count;       // 0x006C (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._GFxDatas_Shops_TA__OnPurchase_0x1");
}

return uClassPointer;
};
```

```
};

void __GFxDATA_Shops_TA__OnPurchase_0x1(TArray<struct FOnlineProductData> Products,
TArray<struct FCurrency> Currencies);
};

// Class TAGame.__GFxDATA_Shops_TA__SetShopViewed_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GFxDATA_Shops_TA__SetShopViewed_0x1 : public UObject
{
public:
int32_t InShopID; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_Shops_TA__SetShopViewed_0x1");
}

return uClassPointer;
};

bool __GFxDATA_Shops_TA__SetShopViewed_0x1(class UGFxDATA_ShopCatalogue_TA* S);
};

// Class TAGame.__GFxDATA_SpecialEvents_TA__HandlePurchaseRewardFromStoreSuccess_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxDATA_SpecialEvents_TA__HandlePurchaseRewardFromStoreSuccess_0x1 : public
UObject
{
public:
class URPC_PurchaseSpecialEventReward_TA* RPC; // 0x0060
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_SpecialEvents_TA__HandlePurchaseRewardFromStoreSuccess_0x1");
}

return uClassPointer;
};
```

```

void __GFxData_SpecialEvents_TA_HandlePurchaseRewardFromStoreSuccess_0x1(class UGFxData_SpecialEventStore_TA* S);
};

// Class TAGame.RPC_SpecialEventBase_TA
// 0x0000 (0x00E8 - 0x00E8)
class URPC_SpecialEventBase_TA : public URPC_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_SpecialEventBase_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_PurchaseSpecialEventReward_TA
// 0x0098 (0x00E8 - 0x0180)
class URPC_PurchaseSpecialEventReward_TA : public URPC_SpecialEventBase_TA
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t                      StoreProductID;     // 0x0130 (0x0004)
[0x0001000000000000]
struct FRocketPassInfo        RocketPassInfo;    // 0x0134 (0x000C)
[0x0001000000000000]
TArray<struct FOnlineProductData> ProductData;   // 0x0140
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineXPReward> RewardDrops;       // 0x0150
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FCurrency>      CurrencyData;      // 0x0160 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FCurrency>      WalletCurrencies; // 0x0170 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_PurchaseSpecialEventReward_TA");
}
}

```

```

return uClassPointer;
};

class URPC_PurchaseSpecialEventReward_TA* SetStoreProductID(int32_t InProductID);
class URPC_PurchaseSpecialEventReward_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.GFxData_SpecialEventStore_TA
// 0x0024 (0x0094 - 0x00B8)
class UGFxData_SpecialEventStore_TA : public UGFxDataRow_X
{
public:
int32_t EventID; // 0x0098 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
TArray<struct FGfxSpecialEventReward> Rewards; // 0x00A0
(0x0010) [0x0001000000402000] (CPF>Transient | CPF_NeedCtorLink)
class UOnlineProductStoreSet_TA* SpecialEventsSet; // 0x00B0
(0x0008) [0x000100004082008] (CPF>ExportObject | CPF>Transient | CPF_Component | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SpecialEventStore_TA");
}

return uClassPointer;
};

void HandlePurchasedRewards(class URPC_PurchaseSpecialEventReward_TA* RPC);
TArray<struct FGfxSpecialEventReward> GetRewards();
struct FProductHashID GetRewardHashID(struct FSpecialEventReward& InReward);
struct FGfxSpecialEventReward CreateEventReward(struct FSpecialEventReward InReward);
void Init(struct FSpecialEventStore StoreData);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_SpecialEvents_TA
// 0x0034 (0x0098 - 0x00CC)
class UGFxData_SpecialEvents_TA : public UGFxDataSingleton_X
{
public:
class USpecialEventConfig_TA* EventData; // 0x0098 (0x0008)
[0x0001000000002000] (CPF>Transient)
uint8_t EventState; // 0x00A0 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t Currency; // 0x00A4 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)

```

```

int32_t SecondsRemaining; // 0x00A8 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
TArray<class UGFxData_SpecialEventStore_TA*> EventStores; // 0x00B0
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UWallet_TA* Wallet; // 0x00C0 (0x0008)
[0x0001000000002000] (CPF_Transient)
unsigned long bSpecialEventButtonVisible : 1; // 0x00C8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SpecialEvents_TA");
}

return uClassPointer;
};

void __GFxData_SpecialEvents_TA__OnShellSet_0x3(class UPsyNetConnection_X* _);
void __GFxData_SpecialEvents_TA__OnShellSet_0x1(class UWallet_TA* InWallet);
void __GFxData_SpecialEvents_TA__OnShellSet_0x2(class UWallet_TA* W, int32_t Id);
void __GFxData_SpecialEvents_TA__GetStoreRewards_0x2(class
URPC_GetSpecialEventCurrencyRewards_TA* RPC);
int32_t __GFxData_SpecialEvents_TA__GetStoreRewards_0x1(struct FSpecialEventStoreConfigs
S);
void __GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x4(class
URPC_PurchaseSpecialEventReward_TA* RPC);
int32_t GetCost(int32_t StoreProductID);
void OnPurchaseSuccess();
void HandlePurchaseRewardFromStoreSuccess(class URPC_PurchaseSpecialEventReward_TA*
RPC);
void OnPurchaseFail();
void HandlePurchaseFail(class URPC_X* RPC);
void PurchaseRewardFromStore(int32_t StoreProductID);
int32_t GetStoreIndex(int32_t InEventID);
void OnStoreRewardsReceived();
void SetStoreRewards(TArray<struct FSpecialEventStore>& InStores);
void GetStoreRewards();
void GetCurrency();
void UpdateSpecialEventButtonVisibility();
void UpdateEventStatus();
void HandleEventConfigChanged(class USpecialEventConfig_TA* InEventData);
void eventOnShellSet();
};

// Class TAGame.__GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x1 : public UObject
{
public:

```

```

class USaveData_TA*           SaveData;          // 0x0060 (0x0008)
[0x0001000000000000]
class UWallet_TA*            WalletReference; // 0x0068 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x1");
}

return uClassPointer;
};

void __GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x3(class
URPC_PurchaseSpecialEventReward_TA* RPC);
void __GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x2(class
URPC_PurchaseSpecialEventReward_TA* RPC);
void __GFxData_SpecialEvents_TA__PurchaseRewardFromStore_0x1(class
URPC_PurchaseSpecialEventReward_TA* RPC);
};

// Class TAGame.Wallet_TA
// 0x00A0 (0x0060 - 0x0100)
class UWallet_TA : public UObject
{
public:
TArray<struct FCurrency>          Currencies;      // 0x0060 (0x0010)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UOnlineProductStoreSet_TA*    CurrencyProductSet; // 0x0070
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
int32_t                            RocketBucksCurrencyID; // 0x0078 (0x0004)
[0x0000000000000002] (CPF_Const)
class USpecialEventConfig_TA*       SpecialEvent;    // 0x0080 (0x0008)
[0x0000800000000001] (CPF_Edit)
struct FScriptDelegate              _EventWalletUpdated__Delegate; // 0x0088
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              _EventWalletCurrencyUpdated__Delegate; // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              _EventWalletCurrencyDataUpdated__Delegate; //
0x00B8 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              _EventReceivedCurrencyDrop__Delegate; // 0x00D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate              _EventReceivedCrewCurrencyDrop__Delegate; //
0x00E8 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Wallet_TA");
}

return uClassPointer;
};

void __Wallet_TA__SetCurrencyTradeholds_0x1(struct FCurrency C);
void __Wallet_TA__OnSpecialEventSet_0x2();
void __Wallet_TA__OnSpecialEventSet_0x1();
void GiveCurrencyDrops(unsigned long bIsCrewRefund, class FString Message, TArray<struct FCurrency>& Drops);
void GiveCurrencyDrop(unsigned long bIsCrewRefund, class FString Message, struct FCurrency& Drop);
void OnSpecialEventSet();
void UpdateWalletCurrencies(TArray<struct FCurrency>& InCurrencies);
void UpdateCurrencyImage(int32_t InCurrencyID, class UTexture* SmallImage, class UTexture* LargeImage);
void UpdateCurrencyName(int32_t InCurrencyID, class FString LocalizedName);
void HandleCurrencyAddedNotification(class UPsyNetService_CurrencyAdded_TA* Service);
void DecreaseCurrency(int32_t InCurrencyID, int32_t Amount);
void IncreaseCurrency(int32_t InCurrencyID, int32_t Amount);
void SetCurrencyTradehold(int32_t InCurrencyID, int32_t InTradehold);
struct FCurrency GetCurrency(int32_t InCurrencyID);
void SetCurrencyTradeholds(TArray<struct FCurrency>& InCurrencies);
void SetCurrencyTotals(TArray<struct FCurrency>& InCurrencies);
void HandleGetWallet(class URPC_GetWallet_TA* RPC);
void HandleOnlinePlayerRemoved(class UOnlinePlayer_X* InOnlinePlayer);
void HandlePerConConnected(class UPsyNetConnection_X* InPsyNetConnection);
void GetWallet();
void GetWalletDelayed();
void Init();
void EventReceivedCrewCurrencyDrop(class UWallet_TA* Wallet, class FString Message, struct FCurrency& Drop);
void EventReceivedCurrencyDrop(class UWallet_TA* Wallet, class FString Message, struct FCurrency& Drop);
void EventWalletCurrencyDataUpdated(class UWallet_TA* Wallet, int32_t CurrencyID);
void EventWalletCurrencyUpdated(class UWallet_TA* Wallet, int32_t CurrencyID);
void EventWalletUpdated(class UWallet_TA* Wallet);
};

// Class TAGame.__GFxData_StartMenu_TA__HandleSyncOnlinePlayerStorageStarted_0x2
// 0x0008 (0x0060 - 0x0068)
class U_GFxData_StartMenu_TA__HandleSyncOnlinePlayerStorageStarted_0x2 : public UObject
{
public:
class UOnlineStorageSyncManager_TA*           OnlineStorageSyncManager;          //
0x0060 (0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_StartMenu_TA__HandleSyncOnlinePlayerStorageStarted_0x2");
}

return uClassPointer;
};

void __GFxData_StartMenu_TA__HandleSyncOnlinePlayerStorageStarted_0x2();
};

// Class TAGame.OnlineStorageSyncManager_TA
// 0x0060 (0x0060 - 0x00C0)
class UOnlineStorageSyncManager_TA : public UObject
{
public:
class UOnlineStorageConfig_TA*           Config;           // 0x0060 (0x0008)
[0x0000800000000001] (CPF_Edit)
class UEOSOnlineStorageSync_TA*          EOSOnlineStorageSync; // 0x0068
(0x0008) [0x0000000000000000]
unsigned long                           bDebugSyncManager : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long                           bSyncInProgress : 1; // 0x0070 (0x0004)
[0x0000000000000000] [0x00000002]
struct FScriptDelegate                 _EventSyncOnlinePlayerStorageStarted__Delegate;//
0x0078 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventSyncSuccess__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _EventSyncComplete__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineStorageSyncManager_TA");
}

return uClassPointer;
};

void __OnlineStorageSyncManager_TA__HandlePsyNetLogin_0x1(class
USaveGameManager_TA* Manager, class USaveData_TA* SaveData, class UError* Error);
void __OnlineStorageSyncManager_TA__SyncCloudData_0x2(class UError* _);
void __OnlineStorageSyncManager_TA__SyncOnlinePlayerStorage_0x1(class UError* E);
void OnlineStorageSyncSuccess(class UObject* RemoteObject);
bool IsOnlinePlayerStorageSyncSuccess();

```

```

void NotifyWhenSyncComplete(struct FScriptDelegate Callback);
void HandleStorageSyncSuccess();
class UAsyncTask* SyncOnlinePlayerStorage();
void RetrySyncOnlinePlayerStorage();
void NotifyOnSyncOnlinePlayerStorageStarted(struct FScriptDelegate Callback);
void SyncCloudData(class UOnlinePlayer_X* OnlinePlayer);
void HandlePsyNetLogin(class UPsyNetConnection_X* C);
void Unload();
void eventConstruct();
void EventSyncComplete();
void EventSyncSuccess(class UOnlineStorageSyncManager_TA* OnlineStorageSyncManager);
void EventSyncOnlinePlayerStorageStarted(class UOnlineStorageSyncManager_TA*
OnlineStorageSyncManager, class UAsyncTask* SyncTask);
};

// Class TAGame.GFxData_StartMenu_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_StartMenu_TA : public UGFxDataSingleton_X
{
public:
class UOnlineGame_X*           OnlineGame;           // 0x0098 (0x0008)
[0x0000800000000001] (CPF_Edit)
class FString                 SigningInText;       // 0x00A0 (0x0010)
[0x000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UGFxModal_X*            SignInModal;         // 0x00B0 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t                      ClickedControllerIndex; // 0x00B8 (0x0004)
[0x0000004000002000] (CPF_Transient)
struct FScriptDelegate        __EventTitleScreenClicked__Delegate; // 0x00C0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_StartMenu_TA");
}

return uClassPointer;
};

void __GFxData_StartMenu_TA__SyncOnlineData_0x1(class USaveGameManager_TA* Manager,
class USaveData_TA* SaveData, class UError* Error);
void __GFxData_StartMenu_TA__HandleSyncOnlinePlayerStorageStarted_0x1(class UError* X);
void PrimaryPlayerSet();
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA* SaveData, class UError* Error);
void OnPrimaryPlayerSet(int32_t ControllerIndex);
class ULocalPlayer_TA* GetPrimaryLocalPlayer();
void OnContinueWithNoUser(class UGFxModal_X* Modal);
void OnShowAccountPicker(class UGFxModal_X* Modal);

```

```

void OnLoginCancelled();
void HandleLoginCancelled();
void HandlePrimaryPlayerLoginFailed(uint8_t PlayerNum, uint8_t ErrorCode);
void HandlePrimaryPlayerLoginChange(uint8_t PlayerNum);
void ClearPrimaryPlayerLoginDelegates(uint8_t PlayerNum);
void PrimaryPlayerSetup(int32_t ControllerIndex);
void OnChoosePlayerName();
void HandleSyncOnlinePlayerStorageStarted(class UOnlineStorageSyncManager_TA*
OnlineStorageSyncManager, class UAsyncTask* SyncTask);
void CreateOnlinePlayerStorageSyncTaskAndModal(class USaveData_TA* SaveData);
bool SyncOnlineData();
void ProgressToMainMenu();
void TitleScreenClicked(int32_t ControllerIndex);
void OnKonamiCode();
void eventOnRemoved();
void eventOnShellSet();
void EventTitleScreenClicked(class UGFxData_StartMenu_TA* StartMenu);
};

// Class TAGame.__GFxData_Stats_TA__SetupStatData_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Stats_TA__SetupStatData_0x1 : public UObject
{
public:
class UStatEvent_TA*           EventStat;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_Stats_TA__SetupStatData_0x1");
}

return uClassPointer;
};

bool __GFxData_Stats_TA__SetupStatData_0x1(struct FName E);
};

// Class TAGame.GFxData_Stats_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_Stats_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFxStatData>      Events;          // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<class UStatEvent_TA*>      StatEvents;       // 0x00A8 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Stats_TA");
}

return uClassPointer;
};

bool __GFxData_Stats_TA__GetExcludedStats_0x2(class UGameMode_TA* M);
bool __GFxData_Stats_TA__GetExcludedStats_0x1(class UGameMode_TA* M);
void LoadSavedStats();
void GetExcludedStats(TArray<struct FName>& Excluded);
void SetupStatData();
void eventOnShellSet();
};

// Class TAGame.__GFxData_System_TA__OnShellSet_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_System_TA__OnShellSet_0x1 : public UObject
{
public:
class UEngineShare_X* EngineShare; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxData_System_TA__OnShellSet_0x1");
}

return uClassPointer;
};

void __GFxData_System_TA__OnShellSet_0x1();

// Class TAGame.GFxData_System_TA
// 0x0080 (0x0098 - 0x0118)
class UGFxData_System_TA : public UGFxDataSingleton_X
{
public:
class FString Language; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString UIState; // 0x00A8 (0x0010)
[0x0000004040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString SharedUIState; // 0x00B8 (0x0010)
};

```

```

[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t LocalPlayerCount; // 0x00C8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
unsigned long bSaveDataLoaded : 1; // 0x00CC (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long bLoggedInPsyNet : 1; // 0x00CC (0x0004)
[0x0000000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long bLoggedInPlatform : 1; // 0x00CC (0x0004)
[0x0000000040000000] [0x00000004] (CPF>EditInlineNotify)
unsigned long bOnline : 1; // 0x00CC (0x0004)
[0x0000000040000000] [0x00000008] (CPF>EditInlineNotify)
unsigned long bUndocked : 1; // 0x00CC (0x0004)
[0x0000000040000000] [0x00000010] (CPF>EditInlineNotify)
uint8_t SplitScreenType; // 0x00D0 (0x0001)
[0x0000000040000000] (CPF>EditInlineNotify)
uint8_t AvailableGamepadType; // 0x00D1 (0x0001)
[0x0000000040000000] (CPF>EditInlineNotify)
float SafeZoneRatio; // 0x00D4 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
float UIScale; // 0x00D8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString Platform; // 0x00E0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString SharedUIState_Unlock; // 0x00F0 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FScriptDelegate __EventSharedUIStateChanged__Delegate; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_System_TA");
}

return uClassPointer;
};

void __GFxData_System_TA__OnShellSet_0x2(class FString _);
bool IsItemUnlockPopupActive();
bool IsTableInitialized(struct FName InTableName);
void SetUIState(class FString NewState);
void HandleSafeZoneRatioUpdated(class UGameViewportClient_TA* GVC);
void HandleUIScaleModifierUpdated(class UGameViewportClient_TA* GVC);
void HandleAvailableGamepadTypeChanged(class UGFxEngine_X* Engine);
void HandleLocalPlayerChanged(class ULocalPlayer* NewPlayer);
void HandleSplitScreenTypeChanged(class UGameViewportClient_TA* GVC);
void SetExclusiveInput(unsigned long bEnabled);
void SetSharedUIState(class FString NewSharedUIState);
void HandleProfileSet(class ULocalPlayer_TA* InPlayer);
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*

```

```

SaveData, class UError* Error);
void HandleLocalPlayerLoginStatusChanged(class UOnlinePlayer_X* InPlayer);
void HandleQuitGame(class UGFxModal_X* Modal);
void HandleLoginResult(class UOnlinePlayerAuthentication_X* Auth);
void HandleOnlineStatusChanged(unsigned long bLinkConnected);
void HandlePsyWebLoginChanged(class UOnlinePlayerAuthentication_X* Auth);
void eventOnRemoved();
void eventOnShellSet();
void EventSharedUIStateChanged();
};

// Class TAGame.__GFxData_TourBracket_TA__InitTeams_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_GFxData_TourBracket_TA__InitTeams_0x1 : public UObject
{
public:
    struct FTourTeam           Team;           // 0x0060 (0x0048)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_TourBracket_TA__InitTeams_0x1");
        }

        return uClassPointer;
    };

    bool __GFxData_TourBracket_TA__InitTeams_0x1(class UGFxData_TourBracketTeam_TA* T);
};

// Class TAGame.GFxData_TourBracketTeam_TA
// 0x0024 (0x0094 - 0x00B8)
class UGFxData_TourBracketTeam_TA : public UGFxDataRow_X
{
public:
    uint64_t           TeamID;           // 0x0098 (0x0008)
    [0x0001000040000000] (CPF_EditInlineNotify)
    class FString      TeamName;         // 0x00A0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    int32_t            TeamColorID;       // 0x00B0 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    int32_t            CustomColorID;     // 0x00B4 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    };
};

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourBracketTeam_TA");
}

return uClassPointer;
};

void __GFxData_TourBracketTeam_TA_SetTeam_0x1(class FString Original, class FString
Sanitized);
void SetTeam(struct FTourTeam& Team);
};

// Class TAGame.GFxData_TourBracket_TA
// 0x0038 (0x0098 - 0x00D0)
class UGFxData_TourBracket_TA : public UGFxDataSingleton_X
{
public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x0098
(0x0008) [0x0001800000000001] (CPF_Edit)
uint64_t                                PlayerTeamID;        // 0x00A0 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                                 TotalRounds;        // 0x00A8 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
uint64_t                                Id;                  // 0x00B0 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
TArray<struct FShopCurrencyInfo>          CurrentPayouts;    // 0x00B8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UTourBracketSync_TA*                BracketSync;        // 0x00C8 (0x0008)
[0x0001000044000009] (CPF>Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourBracket_TA");
}

return uClassPointer;
};

void __GFxData_TourBracket_TA_Show_0x1(class UError* Error);
bool __GFxData_TourBracket_TA_SetBracket_0x1(struct FTourTeam Team);
void HandleDependentMatchUpdated(class UTourEventMatch_TA* TourEventMatch, struct
FTourMatch& Match);
void HandleStartTimeElapsed(class UTourEvent_TA* InTourEvent);
void HandleTourActivated(class UOnlineGameTournaments_TA* InTournaments, class
UTourEvent_TA* TourEvent);
void InitLeaderboard(struct FTourLeaderboard& Leaderboard);
void InitMatches(TArray<struct FTourMatch>& Matches);

```

```

void InitTeams(TArray<struct FTourTeam>& Teams);
void SetBracket(struct FTourBracket& Bracket);
void OnBracketRetrieved();
void UpdateBracket(uint64_t TournamentID);
void hide();
void HandleShowBracketComplete(struct FTourBracket& Bracket);
void HandleTournamentSwapped(uint64_t OldTournamentID, uint64_t NewTournamentID);
void Show(uint64_t TournamentID);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.__GFxData_TourBracketMatchDetails_TA__HandleGetMatchDetails_0x1
// 0x0028 (0x0060 - 0x0088)
class U__GFxData_TourBracketMatchDetails_TA__HandleGetMatchDetails_0x1 : public UObject
{
public:
    struct FTourMatchTeamDetails           Details;           // 0x0060 (0x0028)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_TourBracketMatchDetails_TA__HandleGetMatchDetails_0x1");
        }
    }

    return uClassPointer;
};

bool __GFxData_TourBracketMatchDetails_TA__HandleGetMatchDetails_0x1(class
UGFxData_TourBracketMatchTeamDetails_TA* T);
};

// Class TAGame.GFxData_TourBracketMatchTeamDetails_TA
// 0x002C (0x0094 - 0x00C0)
class UGFxData_TourBracketMatchTeamDetails_TA : public UGFxDataRow_X
{
public:
    uint64_t           TeamID;           // 0x0098 (0x0008)
    [0x0001000040000000] (CPF_EditInlineNotify)
    TArray<struct FTeamGameStats>       GameStats;        // 0x00A0
    (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
    TArray<struct FTourMatchPlayerStats> PlayerStats;      // 0x00B0
    (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
};

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_TourBracketMatchTeamDetails_TA");
}

return uClassPointer;
};

struct FTeamGameStats __GFxData_TourBracketMatchTeamDetails_TA_SetDetails_0x1(int32_t
G);
TArray<struct FTourMatchPlayerStats> GetPlayerStats();
TArray<struct FTeamGameStats> GetTeamGameStats();
void SetDetails(struct FTourMatchTeamDetails InDetails);
};

// Class TAGame.GFxData_TourBracketMatchDetails_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_TourBracketMatchDetails_TA : public UGFxDataSingleton_X
{
public:
uint64_t TournamentID; // 0x0098 (0x0008)
[0x0001000000000000]
int32_t MatchID; // 0x00A0 (0x0004)
[0x0001000000000000]
float PollIntervalSeconds; // 0x00A4 (0x0004)
[0x0001000000000000]
class UAsyncTask* GetMatchDetailsTask; // 0x00A8 (0x0008)
[0x0001000000002000] (CPF_Transient)
class UOnlineGameTournaments_TA* Tournaments; // 0x00B0
(0x0008) [0x0001800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourBracketMatchDetails_TA");
}

return uClassPointer;
};

void __GFxData_TourBracketMatchDetails_TA_SendRequest_0x1(class UError* _);
void OnMatchDetailsRefreshed();
void HandleGetMatchDetails(TArray<struct FTourMatchTeamDetails>& TeamDetails);
void SendRequest();
void DeactivateMatchDetails();
void StartSynchDetails();
void HideMatchDetails();
void ShowMatchDetails(uint64_t InTournamentID, int32_t InMatchID);

```

```

};

// Class TAGame.__GFxData_TourCheckIn_TA__HandleCheckInError_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_TourCheckIn_TA_HandleCheckInError_0x1 : public UObject
{
public:
    uint64_t TournamentID; // 0x0060 (0x0008)
    [0x0010000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_TourCheckIn_TA_HandleCheckInError_0x1");
        }

        return uClassPointer;
    };

    void __GFxDATA_TourCheckIn_TA_HandleCheckInError_0x1(class UGFxModal_X* _);
};

// Class TAGame.GFxData_TourCheckIn_TA
// 0x0060 (0x0098 - 0x00F8)
class UGFxDATA_TourCheckIn_TA : public UGFxDataSingleton_X
{
public:
    class UOnlineGameTournaments_TA* Tournaments; // 0x0098
    (0x0008) [0x00180000000001] (CPF_Edit)
    class FString TeamName; // 0x00A0 (0x0010)
    [0x001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    struct FCustomMatchTeamSettings TeamSettings; // 0x00B0
    (0x0020) [0x001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    class UTournamentSettingsSave_TA* SaveData; // 0x00D0
    (0x0008) [0x001004000002000] (CPF_Transient)
    class FString TournamentJoinTitle; // 0x00D8 (0x0010)
    [0x001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString MatchmakingCancelledOnCheckIn; // 0x00E8
    (0x0010) [0x001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourCheckIn_TA");
        }
    }
}

```

```

return uClassPointer;
};

class UAsyncTask* GetJoiningTask();
void SetCustomColorID(int32_t ColorID);
void SetTeamColorID(int32_t ColorID);
void SetTeamName(class FString InTeamName);
void HandleMatchmakingOnCheckin();
void HandleCheckInError(class UError* InError, uint64_t TournamentID, unsigned long bPromptUnsubscribe);
void HandleRegisterOnCheck(class UTourSettings_TA* Settings);
void PerformCheckIn(class UTourSettings_TA* Settings, unsigned long bPromptUnsubscribeOnFailure);
void HandleRegisterSuccess(class UTourSettings_TA* Settings, struct FTourPrivateCredentials Credentials);
void HandleEnterCredentials(class FString Password, unsigned long bCanceled, uint64_t TournamentID);
void Register(class UTourSettings_TA* Settings, struct FTourPrivateCredentials Credentials);
void CheckIn(uint64_t TournamentID);
bool AlertOnCheckInError(uint64_t TournamentID);
void HandleSaveDataLoaded(class UTournamentSettingsSave_TA* Save);
void eventOnShellSet();
};

// Class TAGame.OnlineGameTournaments_TA
// 0x00B0 (0x00B0 - 0x0160)
class UOnlineGameTournaments_TA : public UOnline_X
{
public:
    class UTourConfig_TA* [0x0001800000000001] (CPF_Edit) Config; // 0x00B0 (0x0008)
    class UTourEvent_TA* [0x0001000000000001] (CPF_Edit) TourEventArchetype; // 0x00B8 (0x0008)
    class UTourService_TA* [0x0001000000000001] (CPF_Edit) ServiceArchetype; // 0x00C0 (0x0008)
    class UTourService_TA* [0x0001800000000001] (CPF_Edit) Service; // 0x00C8 (0x0008)
    class UTourSubscriptions_TA* [0x0001000000000001] (CPF_Edit) Subscriptions; // 0x00D0 (0x0008)
    class UTourRegistration_TA* [0x0001000000000001] (CPF_Edit) Registration; // 0x00D8 (0x0008)
    class UTourCheckIn_TA* [0x0001000000000001] (CPF_Edit) CheckIn; // 0x00E0 (0x0008)
    class UTourCheckInError_TA* [0x0001000000000001] (CPF_Edit) CheckInError; // 0x00E8 (0x0008)
    class UTourPartyCheckIn_TA* [0x0001000000000001] (CPF_Edit) PartyCheckIn; // 0x00F0 (0x0008)
    class UTourPartyLeaderLeave_TA* [0x0008] [0x0001000000000001] (CPF_Edit) PartyLeaderLeave; // 0x00F8
    class UTourCheckInWindowWatcher_TA* [0x0008] [0x0001000000000001] (CPF_Edit) WindowWatcher; // 0x0100
    class UTourRewards_TA* [0x0001000000000001] (CPF_Edit) Rewards; // 0x0108 (0x0008)
};

```

```

class UTourBracketCache_TA*           Brackets;          // 0x0110 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UTourCredentials_TA*           Credentials;       // 0x0118 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UTourSettingsCache_TA*          Cache;            // 0x0120 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UTourMatchServerBrowser_TA*     MatchServerBrowser; // 0x0128
(0x0008) [0x0001000000000001] (CPF_Edit)
class UTourStatus_TA*                Status;            // 0x0130 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UAutoTour_TA*                 AutoTour;          // 0x0138 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UTourEvent_TA*                 TourEvent;         // 0x0140 (0x0008)
[0x0001004000002000] (CPF_Transient)
struct FScriptDelegate              __EventTourActivated__Delegate; // 0x0148
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameTournaments_TA");
}

return uClassPointer;
};

static bool IsTournamentPlaylist(class UGameSettingPlaylist_X* Playlist);
bool IsInTournamentGame(class AGRI_X* GRI);
static void AlertError(class UGFxShell_TA* Shell, class UError* Error, struct FScriptDelegate
Callback, class UClass* ModalClass);
int32_t GetTourRank(struct FUniqueNetId PlayerID, class UTourSettings_TA* Settings);
bool IsTournamentActiveAndLive(uint64_t TournamentID);
bool IsTournamentActive(uint64_t TournamentID);
void HandleTourEventDeactivated(class UTourEvent_TA* InTourEvent, class UError* Error);
bool GetCanRegister(uint64_t TournamentID);
bool IsRegisteredWithCrossPlatformTour();
bool GetIsRegistered();
void NotifyTourActivated(struct FScriptDelegate Callback);
void ActivateTourEvent(struct FUniqueNetId PlayerID, class UTourSettings_TA* Settings, struct
FTourTeam& Team);
void OnNewGame();
void InitServiceProvider();
void OnInit();
void EventTourActivated(class UOnlineGameTournaments_TA* Tournament, class
UTourEvent_TA* Event);
};

// Class TAGame.TourSubscriptions_TA
// 0x0130 (0x0060 - 0x0190)
class UTourSubscriptions_TA : public UObject

```

```

{
public:
class UTourList_TA*           TourList;                      // 0x0060 (0x0008)
[0x0001004004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UGFxTourList_TA*         GFxTourList;                  // 0x0068 (0x0008)
[0x0001000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
TArray<uint64_t>              Created;                     // 0x0070 (0x0010)
[0x0009004000400000] (CPF>NeedCtorLink)
TArray<uint64_t>              AdminOf;                     // 0x0080 (0x0010)
[0x0009004000400000] (CPF>NeedCtorLink)
TArray<uint64_t>              RegisteredTo;                // 0x0090 (0x0010)
[0x0009004000400000] (CPF>NeedCtorLink)
class UTourCheckInWindowWatcher_TA* WindowWatcher;          // 0x00A0
(0x0008) [0x0001004000002000] (CPF>Transient)
class URPC_X*                 GetSubscriptionsRPC;        // 0x00A8 (0x0008)
[0x0001000000002000] (CPF>Transient)
unsigned long                 bSynced : 1;                // 0x00B0 (0x0004)
[0x0001000000002000] [0x00000001] (CPF>Transient)
class UError*                 SyncError;                   // 0x00B8 (0x0008)
[0x0001000000002000] (CPF>Transient)
uint64_t                      LastEarlyLeaveTournamentID; // 0x00C0 (0x0008)
[0x0001004000000000]
uint64_t                      LastSpectatedTourID;       // 0x00C8 (0x0008)
[0x0001004000000000]
struct FScriptDelegate         __EventSubscriptionsSynced__Delegate; // 0x00D0
(0x0018) [0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __EventUnsubscribe__Delegate;     // 0x00E8
(0x0018) [0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __EventSubscribe__Delegate;      // 0x0100
(0x0018) [0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __EventTourCancelled__Delegate; // 0x0118
(0x0018) [0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __EventTournamentSwapped__Delegate; // 0x0130
(0x0018) [0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __Created__ChangeNotify;        // 0x0148 (0x0018)
[0x000100000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __AdminOf__ChangeNotify;        // 0x0160
(0x0018) [0x000100000400000] (CPF>NeedCtorLink)
struct FScriptDelegate         __RegisteredTo__ChangeNotify;   // 0x0178
(0x0018) [0x000100000400000] (CPF>NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourSubscriptions_TA");
}

return uClassPointer;
};

```

```

void __TourSubscriptions_TA__Unsubscribe_0x1(class URPC_TourUnsubscribe_TA* RPC);
void __RegisteredTo__ChangeNotifyFunc();
void __AdminOf__ChangeNotifyFunc();
void __Created__ChangeNotifyFunc();
void SetEarlyLeaveTournament(uint64_t TournamentID);
void SetSpectatedTournament(uint64_t TournamentID);
bool IsSubscribed(uint64_t TournamentID);
bool IsRegistered(uint64_t TournamentID);
bool IsAdmin(uint64_t TournamentID);
void Cancel(uint64_t TournamentID);
void SwapOutActiveTournament(class UTourSettings_TA* NewTourSettings, struct FTourTeam Team);
void HandleKickedFromTournament(class UPsyNetService_TourKicked_TA* Service);
void HandleUnsubscribe(uint64_t TournamentID);
class UAsyncTask* Unsubscribe(struct FUniqueNetId PlayerID, uint64_t TournamentID, unsigned long bUnsubscribeAnyRegisteredTournament);
TArray<class UTourSettings_TA*> Select(TArray<uint64_t> TournamentIDs);
class UTourSettings_TA* GetTournament(uint64_t TournamentID);
void HandleTourActivated(class UOnlineGameTournaments_TA* Tournament, class UTourEvent_TA* TourEvent);
void HandleCheckInWindowChange(class UTourCheckInWindowWatcher_TA* Watcher, class UTourSettings_TA* Tournament);
void UpdateNextCheckInWindow();
void HandleTourRegistered(class UTourSettings_TA* Tournament);
void HandleTourCreated(class UTourSettings_TA* Tournament);
void NotifyWhenSynced(struct FScriptDelegate Callback);
void HandleCrossPlatformChanged();
void AddNewTourResult(struct FTourResult NewResult);
void HandleSubscriptionsSynced(class URPC_TourGetSubscriptions_TA* RPC);
void InitGFxTourList(class UGFxEngine_X* GFxEngine, class UGFxShell_X* Shell);
void SyncTournaments();
static void HandleTournamentCompleted(class UTourSubscriptions_TA* TourSubscription, class UPrimaryAuthLoggedIn_TA* PrimaryAuthLoggedIn, class UEnteredMainMenu_TA* _);
void Init(class UOnlineGameTournaments_TA* Tournament);
void EventTournamentSwapped(uint64_t OldTournamentID, uint64_t NewTournamentID);
void EventTourCancelled(uint64_t TournamentID);
void EventSubscribe(class UTourSubscriptions_TA* Subscriptions, uint64_t TournamentID);
void EventUnsubscribe(class UTourSubscriptions_TA* Subscriptions, uint64_t TournamentID);
void EventSubscriptionsSynced(class UTourSubscriptions_TA* Subscriptions, class UError* Error);
};

// Class TAGame.__GFxDATA_TourCheckIn_TA__PerformCheckIn_0x1
// 0x000C (0x0060 - 0x006C)
class U_GFxDATA_TourCheckIn_TA__PerformCheckIn_0x1 : public UObject
{
public:
    class UTourSettings_TA*           Settings;          // 0x0060 (0x0008)
    [0x0001000000000000]
    unsigned long                     bPromptUnsubscribeOnFailure : 1; // 0x0068
    (0x0004) [0x0001000000000000] [0x00000001]

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_TourCheckIn_TA__PerformCheckIn_0x1");
}

return uClassPointer;
};

void __GFxData_TourCheckIn_TA__PerformCheckIn_0x2();
void __GFxData_TourCheckIn_TA__PerformCheckIn_0x1(class UError* E);
};

// Class TAGame._GFxData_TourCheckIn_TA__Register_0x1
// 0x0028 (0x0060 - 0x0088)
class U_GFxData_TourCheckIn_TA__Register_0x1 : public UObject
{
public:
class UTourSettings_TA*           Settings;          // 0x0060 (0x0008)
[0x0001000000000000]
struct FTourPrivateCredentials     Credentials;      // 0x0068 (0x0020)
[0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_TourCheckIn_TA__Register_0x1");
}

return uClassPointer;
};

void __GFxData_TourCheckIn_TA__Register_0x3(class UError* E);
void __GFxData_TourCheckIn_TA__Register_0x2(class UTourSettings_TA* T);
void __GFxData_TourCheckIn_TA__Register_0x1(class FString P, unsigned long C);
};

// Class TAGame.TourCheckInError_TA
// 0x00D8 (0x0060 - 0x0138)
class UTourCheckInError_TA : public UObject
{
public:
class UOnlineGameTournaments_TA*    Tournaments;      // 0x0060
(0x0008) [0x0001800000000001] (CPF_Edit)
class UOnlineGameParty_X*           Party;           // 0x0068 (0x0008)
[0x0001800000000001] (CPF_Edit)
}

```

```

struct FUniqueNetId           PlayerID;           // 0x0070 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString                 MemberPreCheckInError; // 0x00B8 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UTourSettings_TA*       Settings;          // 0x00C8 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString                 PartyError;        // 0x00D0 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UErrorType*             PartyErrorType;    // 0x00E0 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString                 Error;              // 0x00E8 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UErrorType*             ErrorType;         // 0x00F8 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString                 CrossPlatformNotEnabled; // 0x0100 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                 TournamentRequirementTitle; // 0x0110 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate         __EventTourCheckInError__Delegate; // 0x0120
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourCheckInError_TA");
}

return uClassPointer;
};

void HandleEnableCrossPlatform(class UGFxShell_TA* Shell);
void AlertSignupError(class UGFxShell_TA* Shell, class UError* InError, uint64_t TournamentID,
struct FScriptDelegate Callback);
void HandleMemberPreCheckInErrorMessage(class UOnlineMessageComponent_X*
Component, class UPartyMessage_TourPreCheckInError_TA* Message);
void ClearPartyError();
void HandlePartyDestroyed(class UOnlineGameParty_X* PartyObject);
void UpdatePartyError();
void UpdateError();
void EndErrorCheck();
void BeginErrorCheck(struct FUniqueNetId InPlayerId, uint64_t TournamentID);
void UpdateErrorData(struct FUniqueNetId InPlayerId, uint64_t TournamentID);
void EventTourCheckInError(uint64_t TournamentID, class UError* CheckInError);
};

// Class TAGame.__GFxData_TourCreate_TA__CreateTournament_0x1
// 0x0038 (0x0060 - 0x0098)
class U_GFxData_TourCreate_TA__CreateTournament_0x1 : public UObject
{
public:

```

```

struct FTourCreateSettings           FinalTourSettings;          // 0x0060 (0x0038)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_TourCreate_TA__CreateTournament_0x1");
}

return uClassPointer;
};

void __GFxDATA_TourCreate_TA__CreateTournament_0x1(class UTourSettings_TA* T);
};

// Class TAGame.GFxDATA_OnlineMatch_TA
// 0x007C (0x0098 - 0x0114)
class UGFxDATA_OnlineMatch_TA : public UGFxDATASingleton_X
{
public:
class ULocalPlayer_TA*             TAPlayer;                  // 0x0098 (0x0008)
[0x00000000000000002000] (CPF_Transient)
class UProfile_TA*                Profile;                  // 0x00A0 (0x0008)
[0x00000000000000002000] (CPF_Transient)
class FString                     NoRegionsMessage;        // 0x00A8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                     NotLoggedInPsynetMessage; // 0x00B8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                     BannedMessage;            // 0x00C8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                     NotPartyLeaderMsg;       // 0x00D8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
int32_t                           CountdownSeconds;         // 0x00E8 (0x0004)
[0x00000000000000002000] (CPF_Transient)
class UOnlineGameParty_X*          OnlineParty;              // 0x00F0 (0x0008)
[0x00000000000000002000] (CPF_Transient)
class FString                     OnlineStatusError;        // 0x00F8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UGFxDATA_OnlineMatchStatus_TA* SearchStatusData;        // 0x0108
(0x0008) [0x00000040000002000] (CPF_Transient)
unsigned long                      bSearching : 1;           // 0x0110 (0x0004)
[0x00000080000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_OnlineMatch_TA");
}

return uClassPointer;
};

uint8_t GetStatusOwnerValue();
bool IsStatusOwner();
void ShowMatchmakingCancelOnPlaylistTimeframeExpired();
void HandleMatchmakingError(class UOnlineGameMatchmaking_X* InMatchMaking, class UError* Error);
void ShowMatchmakingCanceledOnPartySizeChangedError();
void HandleMatchmakingCanceledOnPartySizeChanged(class UOnlineGameMatchmaking_X* InMatchMaking);
void HandlePartySizeChanged(class UOnlineGameParty_X* Party, int32_t NewSize, int32_t OldSize);
void HandlePartyChanged(class UOnlineGameParty_X* Party);
void HandleRegionsSynced(class UGFxData_Regions_TA* RegionsData);
void SetError(class FString msg);
void SetOnlineStatusError(class FString NewStatusError);
void UpdateOnlineStatus();
class FString GetOnlineStatusError();
void ClearUpdateOnlineStatusTimer();
void SetUpdateOnlineStatusTimer();
class UOnlineGame_TA* GetOnlineGame();
void OnProfileSet();
void eventOnRemoved();
void eventOnShellSet();
void CancelSearch();
};

// Class TAGame.GFxData_PrivateMatch_TA
// 0x0086 (0x0114 - 0x019A)
class UGFxData_PrivateMatch_TA : public UGFxData_OnlineMatch_TA
{
public:
TArray<struct FGFxRegion>           Regions;          // 0x0118 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString                         ServerName;       // 0x0128 (0x0010)
[0x0000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString                         Password;        // 0x0138 (0x0010)
[0x0000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t                               SelectedMaxPlayerCount; // 0x0148 (0x0004)
[0x0000000040000001] (CPF>Edit | CPF>EditInlineNotify)
unsigned long                         bPublicMatch : 1; // 0x014C (0x0004)
[0x0000000040000001] [0x00000001] (CPF>Edit | CPF>EditInlineNotify)
int32_t                               MinServerNameLength; // 0x0150 (0x0004)
[0x0000000000000003] (CPF>Edit | CPF>Const)
class FString                         AlreadySearchingMessage; // 0x0158 (0x0010)
[0x0000000000408003] (CPF>Edit | CPF>Const | CPF>Localized | CPF>NecCtorLink)
class FString                         AlreadySearchingForPublicMessage; // 0x0168
(0x0010) [0x0000000000408003] (CPF>Edit | CPF>Const | CPF>Localized | CPF>NecCtorLink)
class FString                         ServerNameLengthError; // 0x0178 (0x0010)

```

```

[0x00000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString DuplicateLanServerMessage; // 0x0188 (0x0010)
[0x00000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
uint8_t SearchStatusOwner; // 0x0198 (0x0001)
[0x0000000000002000] (CPF_Transient)
uint8_t SettingsType; // 0x0199 (0x0001)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PrivateMatch_TA");
}

return uClassPointer;
};

uint8_t GetStatusOwnerValue();
void UpdateCustomMatchSettings();
void StartSearchWithSettings(struct FCustomMatchSettings& Settings);
void StartSearch();
void StartClubSearch();
void CancelSearch();
void SetRegionSelection(int32_t Row, unsigned long bSelected);
class FString GetPreferredRegion();
void HandleRegionsSynced(class UGFxData_Regions_TA* RegionsData);
void SetPublic(unsigned long bValue);
void SetPassword(class FString InPassword);
void SetServerName(class FString InServerName);
void SetSelectedMaxPlayerCount(int32_t MaxPlayerCount);
class FString GetCustomMatchRegion();
void SetCustomMatchSettings(struct FCustomMatchSettings& InSettings);
struct FCustomMatchSettings GetCustomMatchSettings();
void eventOnShellSet();
};

// Class TAGame.GFxData_TourCreate_TA
// 0x0086 (0x019A - 0x0220)
class UGFxData_TourCreate_TA : public UGFxData_PrivateMatch_TA
{
public:
class FString PrivateNoPassword; // 0x01A0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString StartTimeTooSoon; // 0x01B0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
int32_t MaxBracketSize; // 0x01C0 (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t RankMin; // 0x01C4 (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t RankMax; // 0x01C8 (0x0004)

```

```

[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t SeriesLength; // 0x01CC (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t FinalSeriesLength; // 0x01D0 (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
TArray<struct FGFxTournamentReward> Rewards; // 0x01D8
(0x0010) [0x0001000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
uint8_t SeedingType; // 0x01E8 (0x0001)
[0x0001000040000001] (CPF_Edit | CPF>EditInlineNotify)
uint8_t TieBreaker; // 0x01E9 (0x0001)
[0x0001000040000001] (CPF_Edit | CPF>EditInlineNotify)
class UOnlineSubsystem* OnlineSub; // 0x01F0 (0x0008)
[0x0001800000000001] (CPF_Edit)
class UOnlineGameTournaments_TA* Tournaments; // 0x01F8
(0x0008) [0x0001800000000001] (CPF_Edit)
class UTourConfig_TA* Config; // 0x0200 (0x0008)
[0x0001800000000001] (CPF_Edit)
class UTourSettings_TA* Settings; // 0x0208 (0x0008)
[0x0001004000002000] (CPF_Transient)
class UTournamentSettingsSave_TA* TournamentSave; // 0x0210
(0x0008) [0x0001000000002000] (CPF_Transient)
class UMapSet_TA* DefaultMapSet; // 0x0218 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourCreate_TA");
}

return uClassPointer;
};

struct FName __GFxData_TourCreate_TA__GetDefaultDisabledMaps_0x2(struct
FGFxMapSelectionData M);
struct FName __GFxData_TourCreate_TA__GetDefaultDisabledMaps_0x1(class UMapData_TA* M);
class FString GetOnlineStatusError();
void HandleMapSelectionChange(class UGFxData_MapSelection_TA* GFxMapSelection, struct
FName MapName, unsigned long bSelected);
void HandleMapListChange(class UGFxData_MapSelection_TA* GFxMapSelection);
void HandlePlatformSelectionChange(class UGFxData_TourPlatforms_TA* GFxPlatforms,
TArray<uint8_t> NewPlatforms);
static void GFxRewardsToTourRewards(TArray<struct FGFxTournamentReward>& GFxRewards,
TArray<struct FTourReward>& TourRewards);
void RemoveReward(int32_t Placement, struct FProductInstanceID InstanceID);
void AddReward(int32_t Placement, struct FProductInstanceID InstanceID);
void SetTieBreaker(uint8_t InTieBreaker);
void SetSeedingType(uint8_t InSeedingType);
void SetFinalSeriesLength(int32_t InFinalSeriesLength);

```

```

void SetSeriesLength(int32_t InSeriesLength);
void SetRankMax(int32_t InRankMax);
void SetRankMin(int32_t InRankMin);
void SetMaxBracketSize(int32_t InMaxBracketSize);
void SetStartTime(struct FDateTimeStruct DateTime);
void HandleError(class UError* Error);
void OnCreateComplete(uint64_t Id);
void HandleCreateTournament(class UTourSettings_TA* Tournament, struct
FTourPrivateCredentials Credentials);
void CreateTournament();
void Update();
TArray<struct FName> GetDefaultDisabledMaps();
class UTourSettings_TA* CreateDefaultSettings();
void OnProfileSet();
void eventOnShellSet();
};

// Class TAGame.__GFxData_TourCreate_TA__GetDefaultDisabledMaps_0x3
// 0x0010 (0x0060 - 0x0070)
class U_GFxDATA_TourCreate_TA__GetDefaultDisabledMaps_0x3 : public UObject
{
public:
TArray<struct FName> DefaultMapNames; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_TourCreate_TA__GetDefaultDisabledMaps_0x3");
}

return uClassPointer;
};

bool __GFxDATA_TourCreate_TA__GetDefaultDisabledMaps_0x3(struct FName MapName);
};

// Class TAGame.__GFxDATA_TourDependentMatch_TA__UpdateDependentMatch_0x1
// 0x0030 (0x0060 - 0x0090)
class U_GFxDATA_TourDependentMatch_TA__UpdateDependentMatch_0x1 : public UObject
{
public:
struct FTourMatch Match; // 0x0060 (0x0030)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_TourDependentMatch_TA__UpdateDependentMatch_0x1");
}

return uClassPointer;
};

void __GFxData_TourDependentMatch_TA__UpdateDependentMatch_0x1(struct FTourBracket B);

// Class TAGame.GFxData_TourBracketMatch_TA
// 0x004C (0x0094 - 0x00E0)
class UGFxData_TourBracketMatch_TA : public UGFxDataRow_X
{
public:
int32_t MatchID; // 0x0098 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t Round; // 0x009C (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t Order; // 0x00A0 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t NextMatch; // 0x00A4 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t PrevMatchA; // 0x00A8 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t PrevMatchB; // 0x00AC (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t TeamA; // 0x00B0 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t TeamB; // 0x00B8 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t TeamAWins; // 0x00C0 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t TeamBWins; // 0x00C4 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t WinnerTeam; // 0x00C8 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
unsigned long bInProgress : 1; // 0x00D0 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bCanSpectate : 1; // 0x00D0 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
class UClass* GFxGameClass; // 0x00D8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourBracketMatch_TA");
}

return uClassPointer;
};

void CreateGames(TArray<struct FTourMatchGame>& Games);
void SetMatch(struct FTourMatch& MatchData);
};

// Class TAGame.GFxData_TourDependentMatch_TA
// 0x0008 (0x00E0 - 0x00E8)
class UGFxData_TourDependentMatch_TA : public UGFxData_TourBracketMatch_TA
{
public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x00E0
(0x0008) [0x0001800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourDependentMatch_TA");
}

return uClassPointer;
};

void HandleRetrievedBracketForDependentMatch(struct FTourBracket& Bracket, struct
FTourMatch& Match);
void UpdateDependentMatch(class UTourEventMatch_TA* TourEventMatch, struct FTourMatch&
Match);
void eventOnRemoved();
};

// Class TAGame.__GFxData_TourEvent_TA__HandleBracketRetrieved_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_TourEvent_TA__HandleBracketRetrieved_0x1 : public UObject
{
public:
uint64_t           OpponentTeamId;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class

```

```

TAGame.__GFxData_TourEvent_TA__HandleBracketRetrieved_0x1");
}

return uClassPointer;
};

bool __GFxData_TourEvent_TA__HandleBracketRetrieved_0x1(struct FTourTeam T);
};

// Class TAGame.GFxData_TourEvent_TA
// 0x0094 (0x0098 - 0x012C)
class UGFxData_TourEvent_TA : public UGFxDataSingleton_X
{
public:
    class UOnlineGameTournaments_TA*           Tournaments;           // 0x0098
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UOnlineGameTourMatchmaking_TA*        TourMatchmaking;      //
    0x00A0 (0x0008) [0x0001800000000001] (CPF_Edit)
    class UTourConfig_TA*                      TourConfig;          // 0x00A8 (0x0008)
    [0x0001800000000001] (CPF_Edit)
    class FString                            Title;                // 0x00B0 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    uint64_t                                Id;                  // 0x00C0 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t                                ScheduleID;         // 0x00C8 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    int32_t                                 SecondsRemaining;   // 0x00D0 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t                                GenerateBracketTime; // 0x00D8 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint8_t                                 Status;              // 0x00E0 (0x0001)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t                                LocalTeamID;        // 0x00E8 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    int32_t                                 Round;               // 0x00F0 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    int32_t                                 TotalRounds;       // 0x00F4 (0x0004)
    [0x0001000040000000] (CPF>EditInlineNotify)
    class FString                           NextOpponent;       // 0x00F8 (0x0010)
    [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
    uint64_t                                EndGameWarningEpochTime; // 0x0108 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t                                EndGameEpochTime;    // 0x0110 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    unsigned long                          bTroubleJoining : 1; // 0x0118 (0x0004)
    [0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
    class UTourSettings_TA*                 ActiveTournament;   // 0x0120 (0x0008)
    [0x000100000002000] (CPF_Transient)
    int32_t                                 MatchID;             // 0x0128 (0x0004)
    [0x000100000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourEvent_TA");
}

return uClassPointer;
};

void __GFxData_TourEvent_TA__OnShellSet_0x3(class UTourCheckInWindowWatcher_TA* W,
class UTourSettings_TA* T);
void __GFxData_TourEvent_TA__OnShellSet_0x2(class UTourList_TA* _);
void __GFxData_TourEvent_TA__OnShellSet_0x1(class UTourSubscriptions_TA* S, uint64_t T);
bool __GFxData_TourEvent_TA__GetNextStartingTournament_0x2(class UTourSettings_TA* T);
int32_t __GFxData_TourEvent_TA__GetNextStartingTournament_0x1(class UTourSettings_TA* L,
class UTourSettings_TA* R);
class UTourSettings_TA* GetNextStartingTournament();
void ReadyForNextMatch();
void ConditionalStepRound();
void HandleFailedToJoinGame(unsigned long bRetrying);
void HandleJoinedGame();
void Deactivate();
void HandleGameDataSelected(int32_t PlaylistId, int32_t MutatorIndex);
void HandleGRISpawned(class AGRI_X* GRI);
void HandleDependentMatchUpdated(class UTourEventMatch_TA* TourEventMatch, struct
FTourMatch& Match);
bool HasBracketDetails();
void HandleBracketRetrieved(struct FTourBracket& Bracket);
void HandleNextMatchReady(class UTourEventMatch_TA* TourEventMatch, struct FTourMatch&
InMatch);
void HandleStartTimeElapsed(class UTourEvent_TA* InTourEvent);
void OnTournamentCancelled();
void HandleTourDeactivated(class UTourEvent_TA* TourEvent, class UError* Error);
void HandleTourActivated(class UOnlineGameTournaments_TA* InTournaments, class
UTourEvent_TA* TourEvent);
uint8_t DetermineStatus();
void QueueUpdate();
void Update();
void Clear();
void eventOnShellSet();
};

// Class TAGame.__GFxData_TourPlatforms_TA__InitPlatformGroups_0x1
// 0x0001 (0x0060 - 0x0061)
class U_GFxData_TourPlatforms_TA__InitPlatformGroups_0x1 : public UObject
{
public:
    uint8_t PlayerPlatform; // 0x0060 (0x0001)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxData_TourPlatforms_TA_InitPlatformGroups_0x1");
}

return uClassPointer;
};

bool __GFxData_TourPlatforms_TA_InitPlatformGroups_0x1(struct FCrossplayGroup G);
};

// Class TAGame.GFxData_TourPlatforms_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_TourPlatforms_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFxTourPlatformGroup> Groups; // 0x0098
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UOnlineSubsystem* OnlineSub; // 0x00A8 (0x0008)
[0x0001800000000001] (CPF_Edit)
struct FScriptDelegate __EventSelectionChange__Delegate; // 0x00B0
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourPlatforms_TA");
}

return uClassPointer;
};

bool __GFxData_TourPlatforms_TA_CreateGFxGroup_0x1(uint8_t P);
bool __GFxData_TourPlatforms_TA_GetSelectedPlatforms_0x1(struct FGFxTourPlatformGroup P);
void SetSelected(int32_t Row, unsigned long bSelected);
void SelectRow(int32_t Row);
int32_t FindPlatformIndex(TArray<uint8_t>& InPlatforms);
static bool CompareOnlinePlatforms(TArray<uint8_t> Right, TArray<uint8_t>& Left);
class FString BuildPlatformLabel(TArray<uint8_t> Group);
TArray<uint8_t> GetSelectedPlatforms();
void SetSelectedPlatforms(TArray<uint8_t>& SelectedPlatforms);
struct FGFxTourPlatformGroup CreateGFxGroup(struct FCrossplayGroup CrossplayGroup);
void InitPlatformGroups(class UCrossplayConfig_X* CrossplayConfig);
void eventOnShellSet();
void EventSelectionChange(class UGFxData_TourPlatforms_TA* GFxPlatforms, TArray<uint8_t> NewPlatforms);

```

```

};

// Class TAGame.__GFxData_TourRegistration_TA__Register_0x2
// 0x0008 (0x0060 - 0x0068)
class U_GFxDATA_TourRegistration_TA__Register_0x2 : public UObject
{
public:
    uint64_t TournamentID; // 0x0060 (0x0008)
    [0x0010000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxDATA_TourRegistration_TA__Register_0x2");
        }

        return uClassPointer;
    };

    void __GFxDATA_TourRegistration_TA__Register_0x2(class UError* E);
};

// Class TAGame.GFxData_TourRegistration_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxDATA_TourRegistration_TA : public UGFxDATA_Singleton_X
{
public:
    class UOnlineGameTournaments_TA* Tournaments; // 0x0098
    (0x0008) [0x00180000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourRegistration_TA");
        }

        return uClassPointer;
    };

    void __GFxDATA_TourRegistration_TA__Register_0x1(class UTourSettings_TA* T);
    void OnRegistrationComplete();
    void HandleError(class UError* Error, uint64_t TournamentID);
    void Register(uint64_t TournamentID);
};

```

```

// Class TAGame.__GFxData_TourSearch_TA__FindPrivate_0x1
// 0x0020 (0x0060 - 0x0080)
class U__GFxData_TourSearch_TA__FindPrivate_0x1 : public UObject
{
public:
    struct FTourPrivateCredentials          Credentials;           // 0x0060 (0x0020)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_TourSearch_TA__FindPrivate_0x1");
        }

        return uClassPointer;
    };

void __GFxData_TourSearch_TA__FindPrivate_0x1(class URPC_TourFindPrivate_TA* RPC);
};

// Class TAGame.RPC_TourFindPrivate_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_TourFindPrivate_TA : public URPC_X
{
public:
    struct FUniqueNetId          PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    struct FTourPrivateCredentials      Credentials;           // 0x0130 (0x0020)
    [0x0001004000400000] (CPF_NeedCtorLink)
    class UTourSettings_TA*          Tournament;           // 0x0150 (0x0008)
    [0x0001004000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourFindPrivate_TA");
        }

        return uClassPointer;
    };

    class URPC_TourFindPrivate_TA* SetCredentials(struct FTourPrivateCredentials& InCredentials);
    class URPC_TourFindPrivate_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

```

```

// Class TAGame.GFxData_TourSearch_TA
// 0x0070 (0x0098 - 0x0108)
class UGFxData_TourSearch_TA : public UGFxDataSingleton_X
{
public:
    class UTourConfig_TA*           Config;           // 0x0098 (0x0008)
    [0x0001800000000000]
    class FString                   Text;             // 0x00A0 (0x0010)
    [0x0001000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
    int32_t                         RankMin;         // 0x00B0 (0x0004)
    [0x0001000040000001] (CPF>Edit | CPF>EditInlineNotify)
    int32_t                         RankMax;         // 0x00B4 (0x0004)
    [0x0001000040000001] (CPF>Edit | CPF>EditInlineNotify)
    int32_t                         GameMode;        // 0x00B8 (0x0004)
    [0x0001000040000001] (CPF>Edit | CPF>EditInlineNotify)
    TArray<struct FGFxRegion>      Regions;          // 0x00C0 (0x0010)
    [0x0001000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
    int32_t                         TeamSize;         // 0x00D0 (0x0004)
    [0x0001000040000001] (CPF>Edit | CPF>EditInlineNotify)
    int32_t                         BracketSize;     // 0x00D4 (0x0004)
    [0x0001000040000001] (CPF>Edit | CPF>EditInlineNotify)
    unsigned long                    bShowIneligibleRank : 1; // 0x00D8 (0x0004)
    [0x0001000040000001] [0x00000001] (CPF>Edit | CPF>EditInlineNotify)
    class UGFxTourList_TA*          GFxTourList;      // 0x00E0 (0x0008)
    [0x000100004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
    class UTourList_TA*            TourList;         // 0x00E8 (0x0008)
    [0x000100004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
    class UProfile_TA*             Profile;          // 0x00F0 (0x0008)
    [0x0001004000002000] (CPF>Transient)
    class UAsyncTask*              SearchTask;       // 0x00F8 (0x0008)
    [0x0001004000002000] (CPF>Transient)
    class UTournamentSettingsSave_TA* TournamentSave; // 0x0100 (0x0008) [0x000100000002000] (CPF>Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourSearch_TA");
        }

        return uClassPointer;
    }

void __GFxData_TourSearch_TA__OnShellSet_0x1(class UGFxData_Regions_TA* R);
void __GFxData_TourSearch_TA__Search_0x1(class URPC_TourGetPublic_TA* RPC);
int32_t __GFxData_TourSearch_TA__HandleSearchResults_0x1(class UTourSettings_TA* L, class UTourSettings_TA* R);
void ResetSearch();
void HandlePrivateSearchResults(class UTourSettings_TA* Tournament, struct FTourPrivateCredentials Credentials);

```

```

void FindPrivate(class FString Title, class FString Password);
void SetShowIneligibleRank(unsigned long bValue);
void SetBracketSize(int32_t InBracketSize);
void SetTeamSize(int32_t InTeamSize);
void SetRankMax(int32_t InRankMax);
void SetRankMin(int32_t InRankMin);
void SetGameMode(int32_t InGameMode);
void SetText(class FString InText);
void SetRegionSelection(int32_t Row, unsigned long bSelected);
void OnSearchComplete();
void HandleSearchResults(TArray<class UTourSettings_TA*>& Results);
void HandleError(class UError* Error);
void Search();
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.__GFxData_TourSubscriptions_TA__Unsubscribe_0x2
// 0x0020 (0x0060 - 0x0080)
class U__GFxData_TourSubscriptions_TA__Unsubscribe_0x2 : public UObject
{
public:
    class UOnlineGameParty_TA*           Party;          // 0x0060 (0x0008)
    [0x0001000000000000]
    struct FScriptDelegate               TimerDelegate; // 0x0068 (0x0018)
    [0x000100000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_TourSubscriptions_TA__Unsubscribe_0x2");
        }

        return uClassPointer;
    }

    void __GFxData_TourSubscriptions_TA__Unsubscribe_0x3();
    void __GFxData_TourSubscriptions_TA__Unsubscribe_0x2();
};

// Class TAGame.GFxData_TourSubscriptions_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_TourSubscriptions_TA : public UGFxDataSingleton_X
{
public:
    class UOnlineGameTournaments_TA*      Tournaments;   // 0x0098
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UTourSubscriptions_TA*          Subscriptions; // 0x00A0 (0x0008)
    [0x0001000000002000] (CPF_Transient)
};

```

```

TArray<struct FSubscriptionInfo>           Created;          // 0x00A8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FSubscriptionInfo>           AdminOf;         // 0x00B8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FSubscriptionInfo>           RegisteredTo;   // 0x00C8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourSubscriptions_TA");
}

return uClassPointer;
};

void __GFxData_TourSubscriptions_TA__Unsubscribe_0x1(class UError* E);
struct FSubscriptionInfo
__GFxData_TourSubscriptions_TA__HandleCreatedChanged_0x1(uint64_t Id);
struct FSubscriptionInfo
__GFxData_TourSubscriptions_TA__HandleAdminOfChanged_0x1(uint64_t Id);
struct FSubscriptionInfo
__GFxData_TourSubscriptions_TA__HandleRegisteredToChanged_0x1(uint64_t Id);
void HandleRegisteredToChanged();
void HandleAdminOfChanged();
void HandleCreatedChanged();
void HandleCancelCreated(class URPC_TourCancelCreated_TA* RPC);
void CancelCreated(uint64_t TournamentID);
void Unsubscribe(uint64_t TournamentID);
void HandleSubscriptionsSynced(class UTourSubscriptions_TA* InSubscriptions, class UError* Error);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.TourErrors_TA
// 0x01A0 (0x0080 - 0x0220)
class UTourErrors_TA : public UErrorList
{
public:
class UErrorType*           TournamentFull;        // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentPaused;      // 0x0088 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentsDisabled; // 0x0090 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           InActiveTournament;   // 0x0098 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           SplitScreenIsInActiveTournament; // 0x00A0
(0x0008) [0x0001000000000002] (CPF_Const)

```

```
class UErrorType*           NoMatchFound;          // 0x00A8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TeamEliminated;        // 0x00B0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           AlreadyRegistered;     // 0x00B8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NotAdmin;              // 0x00C0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NotEligible;          // 0x00C8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           AlreadyStarted;        // 0x00D0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           InvalidTournamentState; // 0x00D8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           InvalidCredentials;    // 0x00E0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentNotFound;     // 0x00E8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentHasConcluded; // 0x00F0
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           CheckInRankIneligible; // 0x00F8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           CheckInNotOpen;         // 0x0100 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           CheckInRequiresPartyLeader; // 0x0108 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TooManyPlayersCheckingIn; // 0x0110 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           CannotMatchmakelfRegistered; // 0x0118
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           CannotCheckInWhileOnline; // 0x0120 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           CannotCancelAfterRegistration; // 0x0128
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentHasBeenCancelled; // 0x0130
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           TeamSizeTooLarge;        // 0x0138 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TeamSizeTooSmall;       // 0x0140 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           PlayerNotRegistered;    // 0x0148 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TeamNameNotAllowed;     // 0x0150 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentNameNotAllowed; // 0x0158
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           TournamentCreateLimitReached; // 0x0160
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           AutoTournamentNoTeamFound; // 0x0168
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           AutoTournamentRequiresCrossplatform; // 0x0170
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           AutoTourmantRequiresCrossplatformNotLeader; //
0x0178 (0x0008) [0x0001000000000002] (CPF_Const)
```

```

class UErrorType*           AutoTournamentBanned;          // 0x0180 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           AutoTournamentResultsRevoked; // 0x0188
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           IneligibleForLateTournament; // 0x0190 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           PlayerNotInTourTeam;          // 0x0198 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           PartyMembersNotInTourTeam;    // 0x01A0
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           KickedNotInPartyTour;         // 0x01A8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           KickedPartyIsCheckingIn;      // 0x01B0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NoSplitscreenDuringCheckin;   // 0x01B8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NoInvitesWhileCheckingIn;     // 0x01C0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NoJoiningWhileCheckingIn;     // 0x01C8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           SkillRequirementsNotMet;       // 0x01D0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           PartySkillRequirementsNotMet; // 0x01D8
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           ScheduleRegionIsLocked;        // 0x01E0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           ScheduleRegionIsInvalid;       // 0x01E8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           ScheduleRegionNotSet;         // 0x01F0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TourCreateCrossplatformDisabled; // 0x01F8
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           CustomTournamentRequiresCrossplatform; // 0x0200
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           CannotSpectateWhileInParty;    // 0x0208 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TourRankDisparity;            // 0x0210 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TourTroubleConnectingToServer; // 0x0218
(0x0008) [0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourErrors_TA");
}

return uClassPointer;
};

```

```

};

// Class TAGame.__GFxData_TradeInFilter_TA__UpdatePossibleTradeInQuantities_0x1
// 0x0010 (0x0060 - 0x0070)
class U_GFxData_TradeInFilter_TA__UpdatePossibleTradeInQuantities_0x1 : public UObject
{
public:
TArray<class UOnlineProduct_TA*>           OnlineProducts;          // 0x0060
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_TradeInFilter_TA__UpdatePossibleTradeInQuantities_0x1");
}

return uClassPointer;
};

void __GFxData_TradeInFilter_TA__UpdatePossibleTradeInQuantities_0x1(class
UGFxData_PossibleTradeIn_TA* P);
};

// Class TAGame.GFxData_PossibleTradeIn_TA
// 0x01C4 (0x0094 - 0x0258)
class UGFxData_PossibleTradeIn_TA : public UGFxDataRow_X
{
public:
struct FProductFilter           Filter;          // 0x0098 (0x0170)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                         TradeInFilterID;    // 0x0208 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
uint8_t                          Quality;         // 0x020C (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                         NumberOfTradeIns; // 0x0210 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                         RemainingItems;   // 0x0214 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
struct FProductTradeInFilter     TradeInFilter;    // 0x0218 (0x0040)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PossibleTradeIn_TA");
}

```

```

}

return uClassPointer;
};

void UpdateTradeInQuantities(TArray<class UOnlineProduct_TA*>& OnlineProducts);
void Init(uint8_t ProductQuality, TArray<class UOnlineProduct_TA*>& OnlineProducts, struct FProductTradeInFilter& InTradeInFilter);
};

// Class TAGame.GFxData_TradeInFilter_TA
// 0x0074 (0x0094 - 0x0108)
class UGFxData_TradeInFilter_TA : public UGFxDataRow_X
{
public:
int32_t Id; // 0x0098 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
class FString Label; // 0x00A0 (0x0010)
[0x0001000040400000] (CPF>NeedCtorLink | CPF>EditInlineNotify)
TArray<class UGFxData_PossibleTradeIn_TA*> PossibleTradeIns; // 0x00B0
(0x0010) [0x0001000004000000] (CPF>NeedCtorLink)
struct FProductTradeInFilter TradeInFilter; // 0x00C0 (0x0040)
[0x0001000000402000] (CPF>Transient | CPF>NeedCtorLink)
class USaveData_TA* SaveData; // 0x0100 (0x0008)
[0x0001000000002000] (CPF>Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TradeInFilter_TA");
}

return uClassPointer;
};

void __GFxData_TradeInFilter_TA__OnRemoved_0x1(class UGFxData_PossibleTradeIn_TA* P);
void __GFxData_TradeInFilter_TA__GenerateTradeInQuantities_0x1(class UGFxData_PossibleTradeIn_TA* P);
void UpdatePossibleTradeInQuantities();
void GenerateTradeInQuantities();
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA* InSaveData, class UError* Error);
void Init(struct FProductTradeInFilter& InTradeInFilter);
void eventOnRemoved();
};

// Class TAGame.__GFxData_Training_TA__HandlePrimaryGameplaySettingsSave_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Training_TA__HandlePrimaryGameplaySettingsSave_0x1 : public UObject
{

```

```

public:
class UGameplaySettingsSave_TA*           Settings;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_Training_TA__HandlePrimaryGameplaySettingsSave_0x1");
}

return uClassPointer;
};

void __GFxDATA_Training_TA__HandlePrimaryGameplaySettingsSave_0x1();
};

// Class TAGame.GFxDATA_Training_TA
// 0x0088 (0x0098 - 0x0120)
class UGFxDATA_Training_TA : public UGFxDATASingleton_X
{
public:
struct FName           TrainingType;      // 0x0098 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                CurrentScore;     // 0x00A0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                CurrentRound;     // 0x00A4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                TotalRounds;      // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString          Difficulty;        // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FName            SelectedFreeplayMap; // 0x00C0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long           bShowBoostMeter : 1; // 0x00C8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long           bShufflePlay : 1;  // 0x00C8 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long           bIsCustomTraining : 1; // 0x00C8 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
TArray<struct FUseAction> UseActions;      // 0x00D0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          TutorialTip;       // 0x00E0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FTrainingRoundProgress> ProgressPerShot;    // 0x00F0
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                CurrentPlaylistIndex; // 0x0100 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t                ControlsVisibility; // 0x0104 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)

```

```
class AGameEvent_TrainingEditor_TA*           TrainingEditor;          // 0x0108
(0x0008) [0x0000800000000000]
class UTrainingProgressTracker_TA*           ProgressTracker;        // 0x0110
(0x0008) [0x0000000000002000] (CPF_Transient)
class UTrainingConfig_TA*                   TrainingConfig;         // 0x0118 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Training_TA");
}

return uClassPointer;
};

void __GFxData_Training_TA__OnShellSet_0x1(class UGameplaySettingsSave_TA* Settings);
void __GFxData_Training_TA__SetTrainingEditorGameEvent_0x1(class UObject* NewSave);
int32_t __GFxData_Training_TA__HandleProgressUpdated_0x3(struct FTrainingRoundProgress RoundProgress);
bool __GFxData_Training_TA__HandleProgressUpdated_0x2(struct FTrainingRoundProgress RoundProgress);
bool __GFxData_Training_TA__HandleProgressUpdated_0x1(struct FTrainingRoundProgress RoundProgress);
bool __GFxData_Training_TA__UpdateDataFromProgress_0x1(struct FTrainingRoundProgress RoundProgress);
bool CanUseTrainingUIV2();
bool CanUseMirroring();
bool CanUseShuffle();
bool CanUsePlaylistOverride();
void SetCurrentPlaylist(struct FName Playlist);
bool HasPerfectedTraining(class FString InType, class FString InDifficulty);
bool HasCompletedTraining(class FString InType, class FString InDifficulty);
class UTrainingEditorMetrics_TA* GetTrainingMetrics();
void ResetTrainingProgress();
void EndTrainingEditorPlaytest();
void OnOpenShotSelectionMenu();
void StartAtPlaylistIndex(int32_t InPlaylistIndex);
void PlayTutorial(unsigned long bAdvanced);
void PlayFreeplayMap(struct FName Map);
void PlayTraining(class FString Type, class FString TrainingDifficulty);
bool UpdateDataFromProgress();
void HandleBoostMeterChanged(class AGameEvent_Tutorial_TA* GameEvent);
class FString ParseDifficulty(uint8_t InDifficulty);
void HandleShufflePlayChanged(unsigned long bNewShufflePlay);
void HandleCurrentPlaylistIndexUpdated(int32_t NewCurrentPlaylistIndex);
void HandleProgressUpdated(class UTrainingPackProgress_TA* UpdatedProgress);
void HandleProgressSet(class UTrainingPackProgress_TA* NewProgress, class UTrainingPackProgress_TA* OldProgress);
void HandleEditorDataChanged(class AGameEvent_TrainingEditor_TA* GameEvent);
```

```

void HandleDifficultyUpdated(class AGameEvent_Tutorial_TA* GameEvent);
void HandleScoreUpdated(class AGameEvent_Tutorial_TA* GameEvent);
void SetTrainingEditorGameEvent(class AGameEvent_TrainingEditor_TA* InGameEvent);
void SetTrainingGameEvent(class AGameEvent_Tutorial_TA* InGameEvent);
void HandleTutorialTipChanged(class AGameEvent_Tutorial_TA* GameEvent, class FString NewTip);
void HandleUseActionsChanged(class AGameEvent_Tutorial_TA* GameEvent);
void HandleTutorialEnded(class UTutorial_TA* Tutorial);
void HandleControlsVisibilityChange(uint8_t Visibility);
void HandlePrimaryGameplaySettingsSave(class UGameplaySettingsSave_TA* Settings);
void HandleMapPrefsSave(class UMapPrefsSave_TA* Save);
void eventOnShellSet();
};

// Class TAGame.__GFxData_Training_TA_SetTrainingEditorGameEvent_0x2
// 0x0008 (0x0060 - 0x0068)
class U__GFxData_Training_TA_SetTrainingEditorGameEvent_0x2 : public UObject
{
public:
    class UTrainingEditorNavigation_TA*           TrainingNavigator;          // 0x0060
    (0x0008) [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxData_Training_TA_SetTrainingEditorGameEvent_0x2");
        }
    }

    return uClassPointer;
};

void __GFxData_Training_TA_SetTrainingEditorGameEvent_0x3();
void __GFxData_Training_TA_SetTrainingEditorGameEvent_0x2();
};

// Class TAGame.TrainingEditorNavigation_TA
// 0x0088 (0x0060 - 0x00E8)
class UTrainingEditorNavigation_TA : public UObject
{
public:
    class UTrainingEditorNavigationConfig_TA*      TrainingNavigationConfig;   // 0x0060
    (0x0008) [0x0001800000000000]
    class UIStartRound_TA*                          RoundStarter_Object;       // 0x0068 (0x0008)
    [0x0001000000000000]
    class UIStartRound_TA*                          RoundStarter_Interface;    // 0x0070 (0x0008)
    [0x0001000000000000]
    class UIGetRoundNumber_TA*                     RoundNumberGetter_Object;  // 0x0078
    (0x0008) [0x0001000000000000]
    class UIGetRoundNumber_TA*                     RoundNumberGetter_Interface; // 0x0080

```

```

(0x0008) [0x0001000000000000]
TArray<int32_t> UnshuffledPlaylist; // 0x0088 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t> ActivePlaylist; // 0x0098 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t CurrentActivePlaylistIndex; // 0x00A8 (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t CurrentUnshuffledPlaylistIndex; // 0x00AC (0x0004)
[0x0009000000002000] (CPF_Transient)
unsigned long bShufflePlay : 1; // 0x00B0 (0x0004)
[0x0009000000002000] [0x00000001] (CPF_Transient)
unsigned long bPlaytestStarted : 1; // 0x00B0 (0x0004)
[0x0001000000002000] [0x00000002] (CPF_Transient)
struct FScriptDelegate __CurrentUnshuffledPlaylistIndex__ChangeNotify;// 0x00B8 (0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __bShufflePlay__ChangeNotify; // 0x00D0 (0x0018) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingEditorNavigation_TA");
}

return uClassPointer;
};

void __TrainingEditorNavigation_TA__InitNavigation_0x2(class AGameEvent_TrainingEditor_TA*__);
void __TrainingEditorNavigation_TA__InitNavigation_0x1(class APlayerController* Caller, class FString CommandString);
void __bShufflePlay__ChangeNotifyFunc();
void __CurrentUnshuffledPlaylistIndex__ChangeNotifyFunc();
bool SetUnshuffledPlaylist(unsigned long bForceSet, TArray<int32_t>& NewUnshuffledPlaylist);
void ShuffleActivePlaylist();
void UpdateActivePlaylist();
bool SetCurrentUnshuffledPlaylistIndex(int32_t NewUnshuffledPlaylistIndex);
bool SetCurrentActivePlaylistIndex(int32_t NewActivePlaylistIndex);
void StartAtUnshuffledPlaylistIndex(int32_t InStartIndex);
void StartAtActivePlaylistIndex(int32_t InStartIndex);
static int32_t GetWrappedPlaylistIndex(int32_t InIndexToWrap, int32_t PlaylistLength);
void HandlePlaytestStarted();
uint8_t HandleGameEventCommand(class FString CommandString);
bool CanUseShuffle();
void InitNavigation(class AGameEvent_TrainingEditor_TA* InTrainingEditor);
};

// Class TAGame.__GFxData_Wallet_TA__GetCurrencyRow_0x1
// 0x0004 (0x0060 - 0x0064)
class U__GFxData_Wallet_TA__GetCurrencyRow_0x1 : public UObject

```

```

{
public:
int32_t InCurrencyID; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxDATA_Wallet_TA__GetCurrencyRow_0x1");
}

return uClassPointer;
};

bool __GFxDATA_Wallet_TA__GetCurrencyRow_0x1(class UGFxDATA_WalletCurrency_TA* C);
};

// Class TAGame.GFxDATA_WalletCurrency_TA
// 0x0034 (0x0094 - 0x00C8)
class UGFxDATA_WalletCurrency_TA : public UGFxDATARow_X
{
public:
int32_t CurrencyID; // 0x0098 (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
uint8_t Type; // 0x009C (0x0001)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
class UTexture* SmallImage; // 0x00A0 (0x0008)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
class UTexture* LargeImage; // 0x00A8 (0x0008)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
int32_t Amount; // 0x00B0 (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
class FString LocalizedName; // 0x00B8 (0x0010)
[0x0001000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxDATA_WalletCurrency_TA");
}

return uClassPointer;
};

void Update(struct FCurrency InCurrency);

```

```

void Init(struct FCurrency InCurrency);
};

// Class TAGame.GFxData_Wallet_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_Wallet_TA : public UGFxDataSingleton_X
{
public:
class UWallet_TA* Wallet; // 0x0098 (0x0008)
[0x0001000000000001] (CPF_Edit)
TArray<class UGFxData_WalletCurrency_TA*> Currencies; // 0x00A0
(0x0010) [0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Wallet_TA");
}

return uClassPointer;
};

void __GFxData_Wallet_TA__HandleWalletUpdated_0x1(struct FCurrency C);
int32_t GetCurrencyIndex(int32_t InCurrencyID);
void OnWalletUpdated();
class UGFxData_WalletCurrency_TA* GetCurrencyRow(int32_t InCurrencyID);
void GetWallet();
void HandleWalletCurrencyDataUpdated(class UWallet_TA* InWallet, int32_t InCurrencyID);
void HandleWalletCurrencyUpdated(class UWallet_TA* InWallet, int32_t InCurrencyID);
void HandleWalletUpdated(class UWallet_TA* InWallet);
void CreateCurrency(struct FCurrency InCurrency);
void OnWalletSet(class UWallet_TA* InWallet);
void eventOnShellSet();
};

// Class TAGame.__GFxEngine_TA__InitOnlineSub_0x1
// 0x0008 (0x0060 - 0x0068)
class U__GFxEngine_TA__InitOnlineSub_0x1 : public UObject
{
public:
class UOnline_X* OnlineX; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.__GFxEngine_TA__InitOnlineSub_0x1");
}

return uClassPointer;
};

void __GFxEngine_TA__InitOnlineSub_0x1();

// Class TAGame.__GFxHUD_TA__GetVoteBySubject_0x1
// 0x0001 (0x0060 - 0x0061)
class U__GFxHUD_TA__GetVoteBySubject_0x1 : public UObject
{
public:
    uint8_t           VoteSubject;           // 0x0060 (0x0001)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__GFxHUD_TA__GetVoteBySubject_0x1");
        }
    }

    return uClassPointer;
};

bool __GFxHUD_TA__GetVoteBySubject_0x1(class AVoteActor_TA* OtherVote);
};

// Class TAGame.HUDBase_TA
// 0x0070 (0x0320 - 0x0390)
class AHUDBase_TA : public AHUD_X
{
public:
    class UGFxShell_TA*           Shell;           // 0x0320 (0x0008)
    [0x000008000002000] (CPF_Transient)
    class FString                 DisconnectedString; // 0x0328 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                 ServerMaintString; // 0x0338 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                 PartyChatTitle; // 0x0348 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class UGFxData_Chat_TA*       ChatData;        // 0x0358 (0x0008)
    [0x000004000002000] (CPF_Transient)
    struct FScriptDelegate        _EventGameMessage__Delegate; // 0x0360
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate        _EventChatMessage__Delegate; // 0x0378
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.HUDBase_TA");
}

return uClassPointer;
};

class FString FormatTimeStamp(int32_t TimeStamp);
void DrawLookupTextures();
void DrawHUD();
void NotifyChatBannedPermanently();
void NotifyChatBanned(uint64_t ChatBanExpiration);
void NotifyChatDisabled(float Time);
void NotifyServerShutdown();
void ToggleScoreboard();
bool ChatWithExternalKeyboard(class FString Title, struct FScriptDelegate Callback);
void OpenPartyChat();
void OpenTeamChat();
void OpenChat();
void ToggleMidGameMenu();
void UpdatePRIData();
void ReceiveMessage(class UMessage_TA* Message, class FString Text);
bool ShouldDeliverMessage(struct FGfxChatMessage& GFxMessage);
void AddChatMessage(struct FGfxChatMessage& GFxMessage);
void OnChatMessage(struct FChatMessage NewMsg);
void EraseChatByType(uint8_t ChatChannel, struct FUniqueNetId Personald);
void RepopulateChatHistory(uint8_t ChatChannel, struct FUniqueNetId Personald);
void LobbyMessage_TA(class FString PlayerName, class FString Message, uint8_t ChatChannel,
class APlayerReplicationInfo* SenderPRI, struct FUniqueNetId SenderId);
void IndividualMessage_TA(class FString PlayerName, class FString Message, uint8_t
ChatChannel, struct FUniqueNetId Personald);
void Message_TA(class APlayerReplicationInfo* PRI, class FString Message, uint8_t
ChatChannel, unsigned long bPreset, int32_t TimeStamp, struct FUniqueNetId Recipient);
class UOnlineGame_X* GetOnlineGame();
class UOnlinePlayer_TA* GetOnlinePlayer();
class AGameEvent_TA* GetGameEvent();
void eventDestroyed();
void Init();
void EventChatMessage(struct FChatMessage Message);
void EventGameMessage(class UMessage_TA* Message, class FString Text);
};

// Class TAGame.GFxHUD_TA
// 0x01B0 (0x0390 - 0x0540)
class AGFxHUD_TA : public AHUDBase_TA
{
public:
class FString UIState; // 0x0390 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)

```

```

class UClass*           GFxPRIClass;           // 0x03A0 (0x0008)
[0x0000000000000000]
class UGFxMoviePlayer_X*      MatchInfoMovieArchetype;       // 0x03A8
(0x0008) [0x0000000000000001] (CPF_Edit)
class UGFxMoviePlayer_X*      MatchInfoMovie;           // 0x03B0 (0x0008)
[0x000008000002000] (CPF_Transient)
class UGFxNameplatesManager_TA*      NameplatesManager;        // 0x03B8
(0x0008) [0x000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class FString            PlayerJoinString;          // 0x03C0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            PlayerLeaveString;         // 0x03D0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            PlayerTimedOutString;       // 0x03E0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            ChatTitle;                 // 0x03F0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            TeamChatTitle;            // 0x0400 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            ChatDisabledMessage;        // 0x0410 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            ReplayQueSaveMessage;       // 0x0420 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            ChatBannedMessage;          // 0x0430 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString            ChatBannedPermanentlyMessage; // 0x0440
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class APRI_TA*           OwnerPRI;                // 0x0450 (0x0008)
[0x000008000002000] (CPF_Transient)
class ACar_TA*           OwnerCar;                // 0x0458 (0x0008)
[0x000008000002000] (CPF_Transient)
class AGameEvent_TA*      GameEvent;               // 0x0460 (0x0008)
[0x000008000002000] (CPF_Transient)
class AGameEvent_Team_TA*      TeamGameEvent;        // 0x0468
(0x0008) [0x000008000002000] (CPF_Transient)
class UGFxData_GameEvent_TA*      GameEventData;        // 0x0470
(0x0008) [0x000008000002000] (CPF_Transient)
class UGFxData_LocalCar_TA*      CarData;                // 0x0478 (0x0008)
[0x000008000002000] (CPF_Transient)
TArray<class UGFxData_PRI_TA*>      PRIData;               // 0x0480 (0x0010)
[0x000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UGFxData_MiniScoreboard_TA*      MiniScoreboard;        // 0x0490
(0x0008) [0x00100000002000] (CPF_Transient)
class UGFxData_EOSGameClipsMaskArea_TA*
EOSGameClipsMaskHandler;           // 0x0498 (0x0008) [0x0001004000002000]
(CPF_Transient)
TArray<class UGFxData_PlayerRecord_TA*>      PlayerRecordData;        // 0x04A0
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long             bAttackTargetActive : 1;           // 0x04B0 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long             bShowedDisconnectMessage : 1;        // 0x04B0
(0x0004) [0x0000000000002000] [0x00000002] (CPF_Transient)
struct FVector            AttackTargetWorldPosition;       // 0x04B4 (0x000C)
[0x0000000000000000]

```

```

struct FVector                               AttackTargetPositionOffset;           // 0x04C0 (0x000C)
[0x0000000000000000]
float                                         CloseScoreboardTime;                // 0x04CC (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<class APRI_TA*>                     PendingShowAddMessagePlayers;      // 0x04D0
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class AActor*                                LastTarget;                      // 0x04E0 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                                         TestLastLogTime;                 // 0x04E8 (0x0004)
[0x0000000000002000] (CPF_Transient)
class ACar_TA*                               CarDisplayingBoost;              // 0x04F0 (0x0008)
[0x0000000000000000]
class UAdHocBeacon_X*                       AdHocBeacon;                   // 0x04F8 (0x0008)
[0x0000800000000000]
class FString                                ServerMigrationTitle;            // 0x0500 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                                ServerMigrationBody;             // 0x0510 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UGFxData_Scoreboard_TA*                 Scoreboard;                    // 0x0520 (0x0008)
[0x0001000000002000] (CPF_Transient)
struct FScriptDelegate                        __VoteLambda__Delegate;          // 0x0528
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_TA");
}

return uClassPointer;
};

void __GFxHUD_TA_HandleServerNameChanged_0x1(class FString O, class FString S);
class FString __GFxHUD_TA_UpdateScoreboardStatNames_0x1(struct FScoreboardStat S);
void __GFxHUD_TA_SetGameEvent_0x1(class AGameEvent_Soccar_TA* Soccar);
int32_t __GFxHUD_TA_UpdatePRIData_0x2(class UGFxData_PRI_TA* A, class UGFxData_PRI_TA* B);
bool __GFxHUD_TA_UpdatePRIData_0x1(class UGFxData_PRI_TA* P);
void SetRespawnTimeRemaining(int32_t NewTime);
void HandleReplaceBot(class APRI_TA* InPRI);
struct FScreenLocation GetScreenLocation(struct FVector WorldLocation, float SafeZone);
void PrintDebugInfo(class UDebugDrawer* Drawer);
void DrawNameplatesTexture();
void DrawHUD();
void HandleReplayAutoSave(class AReplayDirector_TA* Director);
void HandleVoteFinished(class AVoteActor_TA* VoteActor);
class UGFxData_Vote_TA* GetVoteBySubject(uint8_t VoteSubject);
class UGFxData_Vote_TA* GetVoteByRow(int32_t RowIndex);
int32_t GetVoteRow(struct FScriptDelegate MatchCondition);
void HandleVoteStarted(class AVoteActor_TA* VoteActor);

```

```
void NotifyChatBannedPermanently();
void NotifyChatBanned(uint64_t ChatBanExpiration);
void NotifyChatDisabled(float Time);
void ChatPreset(int32_t Index);
void HandleInternetConnectionChanged(unsigned long bConnected);
void NotifyClientTravel();
void NotifyServerMigration();
void NotifyServerShutdown();
void HandlePRICameraChanged(class APRI_TA* InPRI);
void PostMatchRequeue();
void CloseScoreboard();
void OpenScoreboard();
void OpenMidgameMenu();
void ReceiveMessage(class UMessage_TA* Message, class FString Text);
void HandleStatTickerMessage(class APRI_TA* Receiver, class APRI_TA* Victim, class UStatEvent_TA* StatEvent);
bool CanDisplayStatEvent(class UStatEvent_TA* StatEvent);
void HandleStatEvent(class APRI_TA* PRI, class UStatEvent_TA* StatEvent, int32_t Count);
void OpenPartyChat();
void OpenTeamChat();
void OpenChat();
bool CanOpenChat(uint8_t InChatFilter);
class UGameplaySettingsSave_TA* GetGameplaySettingsSave();
bool AnyPlayerChatRestricted();
void UpdateCarData();
class ACar_TA* GetFocusCar();
void SetAttackTarget(class AActor* InTarget);
void UpdatePotentialTarget(int32_t Direction);
void UpdateCarAttackTarget(unsigned long bPrevAttackTargetActive);
bool AttackTargetActive(unsigned long bPrevAttackTargetActive);
bool IsNearOffScreen(struct FScreenLocation& ScreenLoc);
float ScreenClamp(float val);
void UpdateCarDataFromCar(class ACar_TA* Car);
void UpdateCameraYaw();
void eventTick(float DeltaTime);
int32_t GetBoostAmountForUI(class ACarComponent_Boost_TA* Boost);
void UpdatePRIData();
void SetOwnerCar(class ACar_TA* NewCar);
void HandlePawnChanged(class APlayerController_X* PC, class APawn* OldPawn, class APawn* NewPawn);
void OnAllTeamsCreated();
void HandleAllTeamsCreated(class AGameEvent_Team_TA* G);
void HandleGameStateChanged(class AGameEvent_TA* G);
void HandlePlayerRemoved(class AGameEvent_TA* G, class APRI_TA* InPRI);
void HandlePlayerNameChanged(class APRI_X* InPRI);
void HandlePlayerIDChanged(class APRI_X* InPRI);
void HideNameplates();
void ShowNameplates();
void ToggleNameplates();
void UpdatePendingAddMessagePlayers();
void HandlePlayerAdded(class AGameEvent_TA* G, class APRI_TA* InPRI);
class FString GetPlayerName(class APRI_TA* PRI);
void UpdateGFxPlayerRecord(class APRI_TA* InPRI);
void UpdateSessionStatus();
```

```

void DestroyGFxPRIData(class UGFxData_PRI_TA*OldData);
class UGFxData_PRI_TA* CreateGFxPRIData();
void HandleReservationsUpdated(class AGRI_X*GRI);
bool IsValidID(struct FUniqueNetId PlayerID);
class UGFxData_PRI_TA* GetPRIDataFromID(struct FUniqueNetId PlayerID);
int32_t GetPRIDataIndex(class APRI_TA*InPRI);
void HandleMatchTypeSet(class AGameEvent_TA*InGameEvent);
void SetGameEvent(class AGameEvent_TA*InGameEvent);
void HandleGameEventChanged(class APRI_TA*InPRI);
void SetTeam(class ATeam_TA*Team);
void HandleTeamChanged(class APRI_X*InPRI);
void UpdateScoreboardStatNames();
void SetOwnerPRI(class APRI_TA*InPRI);
void HandleReceivedPRI(class APlayerController_X*PC);
void UpdateMutatorTags(class FString MatchTags);
void UpdateMutatorTagsFromPreset(class UPresetMutators_X*Presets);
void HandleGameDataSelected(int32_t PlaylistId,int32_t MutatorIndex);
void HandleServerNameChanged(class AGRI_X*GRI);
void HandleCanVoteForfeitChanged(class AGameEvent_TA*InGameEvent);
void HandlePenaltyChanged(class AGameEvent_TA*InGameEvent);
void HandleGRIReplicated(class AGRI_X*GRI);
void HandleHideWorldUIChanged(class UGameShare_TA*G);
void eventDestroyed();
void InitGFxData();
void InitGfx();
void Init();
bool VoteLambda(class AVoteActor_TA*InVoteActor);
};


```

```

// Class TAGame.__GFxHUD_TA__HandlePlayerAdded_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxHUD_TA__HandlePlayerAdded_0x1 : public UObject
{
public:
class APRI_TA* InPRI; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__GFxHUD_TA__HandlePlayerAdded_0x1");
}

return uClassPointer;
};

bool __GFxHUD_TA__HandlePlayerAdded_0x1(class UGFxData_PRI_TA* P);
};

// Class TAGame.__GFxHUD_TA__HandlePlayerNameChanged_0x1

```

```
// 0x0008 (0x0060 - 0x0068)
class U_GFxFxHUD_TA_HandlePlayerNameChanged_0x1 : public UObject
{
public:
class APRI_X* InPRI; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxFxHUD_TA_HandlePlayerNameChanged_0x1");
}

return uClassPointer;
};

bool __GFxFxHUD_TA_HandlePlayerNameChanged_0x1(class UGFxFxData_PRI_TA* P);
};

// Class TAGame._GFxFxHUD_TA_HandleVoteFinished_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxFxHUD_TA_HandleVoteFinished_0x1 : public UObject
{
public:
class AVoteActor_TA* VoteActor; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._GFxFxHUD_TA_HandleVoteFinished_0x1");
}

return uClassPointer;
};

bool __GFxFxHUD_TA_HandleVoteFinished_0x1(class AVoteActor_TA* OtherVote);
};

// Class TAGame._GFxFxModal_Processing_TA_CreateAndBind_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxFxModal_Processing_TA_CreateAndBind_0x1 : public UObject
{
public:
class UAsyncTask* Task; // 0x0060 (0x0008)
```

```
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxModal_Processing_TA__CreateAndBind_0x1");
}

return uClassPointer;
};

void __GFxModal_Processing_TA__CreateAndBind_0x1(class UGFxModal_X* M);
};

// Class TAGame.GFxModal_Processing_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Processing_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Processing_TA");
}

return uClassPointer;
};

static class UGFxModal_X* CreateAndBind(class UGFxShell_TA* Shell, class UAsyncTask* Task);
static class UGFxModal_X* CreateAndBindNoCancel(class UGFxShell_TA* Shell, class
UAsyncTask* Task);
};

// Class TAGame._GFxModal_Processing_TA__CreateAndBindNoCancel_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxModal_Processing_TA__CreateAndBindNoCancel_0x1 : public UObject
{
public:
class UGFxModal_X* Modal; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxModal_Processing_TA__CreateAndBindNoCancel_0x1");
}

return uClassPointer;
};

void __GFxModal_Processing_TA__CreateAndBindNoCancel_0x1();

// Class TAGame._GFxNameplatesManager_TA__HandleCameraStateChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxNameplatesManager_TA__HandleCameraStateChanged_0x1 : public UObject
{
public:
class AActor*           CameraTargetOwner;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxNameplatesManager_TA__HandleCameraStateChanged_0x1");
}

return uClassPointer;
};

bool __GFxNameplatesManager_TA__HandleCameraStateChanged_0x1(class
UGFxData_Nameplate_TA* N);
};

// Class TAGame.GFxData_Nameplate_TA
// 0x0054 (0x0094 - 0x00E8)
class UGFxData_Nameplate_TA : public UGFxDataRow_X
{
public:
float                  HideFullNameplateDistance;    // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                  LivesVisibilityFadeMinDistance; // 0x009C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                  LivesVisibilityFadeMaxDistance; // 0x00A0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class FString           PlayerName;                // 0x00A8 (0x0010)
[0x0000004040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                Team;                     // 0x00B8 (0x0004)

```

```

[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long          bHideFullNameplate : 1;           // 0x00BC (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bIsTargetLocked : 1;           // 0x00BC (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long          bInKnockoutGameMode : 1;        // 0x00BC (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long          bIsDistracted : 1;           // 0x00BC (0x0004)
[0x0001000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long          bLocalPlayer : 1;           // 0x00BC (0x0004)
[0x0000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long          bRenderBoostAmount : 1;        // 0x00BC (0x0004)
[0x0001000040000000] [0x00000020] (CPF_EditInlineNotify)
unsigned long          bShowShortcut : 1;           // 0x00BC (0x0004)
[0x0000004000002000] [0x00000040] (CPF_Transient)
float                 LivesVisibility;           // 0x00C0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t               RemainingLives;           // 0x00C4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t               MaxLives;                // 0x00C8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float                 BoostAmount;              // 0x00CC (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
class UNameplateComponent_TA*      NameplateComponent; // 0x00D0
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UProfileGameplaySave_TA*     GameplaySettings; // 0x00D8
(0x0008) [0x0000004000002000] (CPF_Transient)
class UPersona_TA*                Persona;       // 0x00E0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Nameplate_TA");
}

return uClassPointer;
};

void __GFxData_Nameplate_TA__HandleProfileSet_0x1(class UProfileGameplaySave_TA*
Settings);
void SetShowShortcut(unsigned long bValue);
bool CanRenderBoostNameplate(class APRI_TA* InPRI);
void Tick();
void HandlePersonaUpdated(class UPersona_TA* InPersona);
void HandlePRIDistracted(class APRI_TA* InPRI);
void SetPlayerName(class APRI_TA* InPRI);
class UOnlineGame_X* GetOnlineGame();
void HandlePlayerNameSet(class UNameplateComponent_TA* InNameplateComponent);

```

```

void SetNameplateComponent(class UNameplateComponent_TA* InNameplateComponent);
void HandleProfileSet(class ULocalPlayer_TA* InLocalPlayer);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxNameplatesManager_TA
// 0x0080 (0x0070 - 0x00F0)
class UGFxNameplatesManager_TA : public UComponent
{
public:
class UGFxMoviePlayer_X*           MovieArchetype;           // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UNameplateMeshComponent_TA*   NameplateMeshArchetype; // 0x0078 (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UGFxShell_X*                Shell;                  // 0x0080 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGFxMoviePlayer_X*           Movie;                  // 0x0088 (0x0008)
[0x0000000000002000] (CPF_Transient)
unsigned long                      bHidden : 1;          // 0x0090 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long                      bShowingShortcuts : 1; // 0x0090 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long                      bDebug : 1;           // 0x0090 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
TArray<class UNameplateComponent_TA*> ComponentSlots;      // 0x0098
(0x0010) [0x000000004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
TArray<class UNameplateComponent_TA*> PendingComponents;    // 0x00A8 (0x0010) [0x000000004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
TArray<class UGFxData_Nameplate_TA*> NameplateRows;        // 0x00B8
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UNameplateMeshComponent_TA*> Nameplates;        // 0x00C8 (0x0010) [0x000000004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
struct FScriptDelegate           __EventNameplateSet__Delegate; // 0x00D8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxNameplatesManager_TA");
}

return uClassPointer;
};

void __GFxNameplatesManager_TA__HandleCameraStateChanged_0x2(class

```

```

UGFxData_Nameplate_TA* N);
void DrawDebug(class UNameplateMeshComponent_TA* Nameplate);
void SetShowShortcuts(unsigned long bInShowShortcuts);
void ToggleShowShortcuts();
void Tick();
void UpdateNameplatesVisibility();
void SetHidden(unsigned long bHide);
class UNameplateMeshComponent_TA* GetNameplate(class UNameplateComponent_TA*
InComponent);
class UNameplateMeshComponent_TA* SetNameplate(int32_t Index, class
UNameplateComponent_TA* InComponent);
void SetPlayerData(int32_t Index, class UNameplateComponent_TA* InComponent);
void Update();
class UGameShare_TA* GetGameShare();
void QueUpdate();
void HandleGameStateChanged(class AGameEvent_TA* GameEvent);
void HandleKnockOutGameEventChanged(class APRI_TA* PRI);
void HandleLivesChanged(class APRI_KnockOut_TA* PRI);
void HandlePRITAChanged(class APRI_TA* P);
void HandlePRIXChanged(class APRI_X* P);
void HandleVehicleChanged(class AVehicle_TA* Vehicle);
void HandleCarChanged(class ACar_TA* Car);
void UpdatePendingComponents();
void HandleNameplateRemoved(class UGameShare_TA* G, class UNameplateComponent_TA*
InComponent);
void HandleNameplateAdded(class UGameShare_TA* G, class UNameplateComponent_TA*
InComponent);
void HandleCameraStateChanged(class ACamera_X* Camera, class UCameraState_X*
CameraState);
void Destroyed();
void Init(class UGFxShell_X* InShell);
void EventNameplateSet(class UNameplateMeshComponent_TA* Nameplate);
};

// Class TAGame.__GFxProductAttributeMap_TA__MapAttribute_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxProductAttributeMap_TA__MapAttribute_0x1 : public UObject
{
public:
class UClass* AttributeClass; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__GFxProductAttributeMap_TA__MapAttribute_0x1");
}

return uClassPointer;
}

```

```

};

bool __GFxProductAttributeMap_TA__MapAttribute_0x1(class UClass* GFxClass);
};

// Class TAGame.GFxProductAttributeMap_TA
// 0x0010 (0x0060 - 0x0070)
class UGFxProductAttributeMap_TA : public UObject
{
public:
TArray<class UClass*> GFxClasses; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxProductAttributeMap_TA");
}

return uClassPointer;
};

class UClass* MapAttribute(class UClass* AttributeClass);
TArray<class UClass*> MapAttributes(TArray<class UClass*>& Attributes);
};

// Class TAGame._GFxShell_TA_SetInputActionEnabled_0x1
// 0x000C (0x0060 - 0x006C)
class U_GFxShell_TA_SetInputActionEnabled_0x1 : public UObject
{
public:
struct FName ActionName; // 0x0060 (0x0008)
[0x0000000000000000]
unsigned long bEnabled : 1; // 0x0068 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._GFxShell_TA_SetInputActionEnabled_0x1");
}

return uClassPointer;
};

```

```
void __GFxShell_TA_SetInputActionEnabled_0x2();
void __GFxShell_TA_SetInputActionEnabled_0x1(class UPlayer* P);
};

// Class TAGame.__GFxShell_TA_ShowRankedReconnectModal_0x1
// 0x0070 (0x0060 - 0x00D0)
class U_GFxShell_TA_ShowRankedReconnectModal_0x1 : public UObject
{
public:
    struct FServerReservationData           Reservation;          // 0x0060 (0x0070)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxShell_TA_ShowRankedReconnectModal_0x1");
        }

        return uClassPointer;
    }

void __GFxShell_TA_ShowRankedReconnectModal_0x1(class UGFxModal_X* _);

// Class TAGame.__GFxTourList_TA_HandleTournamentsChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U_GFxTourList_TA_HandleTournamentsChanged_0x1 : public UObject
{
public:
    class UTourList_TA*                 InTourList;          // 0x0060 (0x0008)
    [0x0001000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__GFxTourList_TA_HandleTournamentsChanged_0x1");
        }

        return uClassPointer;
    }

void __GFxTourList_TA_HandleTournamentsChanged_0x1(class UTourSettings_TA* T);
};
```

```

// Class TAGame.TourList_TA
// 0x0068 (0x0070 - 0x00D8)
class UTourList_TA : public UComponent
{
public:
TArray<class UTourSettings_TA*>           Tournaments;           // 0x0070
(0x0010) [0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FTourResult>           Results;           // 0x0080 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventTournamentsChanged__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventTournamentAdded__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventResultsChanged__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourList_TA");
}

return uClassPointer;
};

void __TourList_TA__SetTournaments_0x1(class UTourSettings_TA* T);
struct FTourResult GetTournamentResult(uint64_t TournamentID);
void AddTournamentResult(struct FTourResult InTourResult);
void Remove(uint64_t TournamentID);
void AddUnique(class UTourSettings_TA* Tournament);
void SetResults(TArray<struct FTourResult>& InResults);
void SetTournaments(TArray<class UTourSettings_TA*>& NewTournaments);
void EventResultsChanged(class UTourList_TA* Publisher);
void EventTournamentAdded(class UTourSettings_TA* Tournament);
void EventTournamentsChanged(class UTourList_TA* Publisher);
};

// Class TAGame.GFxData_TourSubscription_TA
// 0x001C (0x0094 - 0x00B0)
class UGFxData_TourSubscription_TA : public UGFxDataRow_X
{
public:
class UTourSettings_TA*           TourSettings;           // 0x0098 (0x0008)
[0x0001000000000000]
class UTourList_TA*           TourList;           // 0x00A0 (0x0008)
[0x000100004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
int32_t           Difficulty;           // 0x00A8 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t           Placement;           // 0x00AC (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourSubscription_TA");
}

return uClassPointer;
};

TArray<int32_t> GetSeriesRoundLengths();
void UpdateTourResult(class UTourList_TA* InTourList);
void SetTourList(class UTourList_TA* InTourList);
void SetProxy(class UObject* InProxyObject);
void eventOnRemoved();
};

// Class TAGame.GFxTourList_TA
// 0x0018 (0x0070 - 0x0088)
class UGFxTourList_TA : public UComponent
{
public:
class UGFxShell_X* Shell; // 0x0070 (0x0008)
[0x0001000000002000] (CPF_Transient)
class UTourList_TA* TourList; // 0x0078 (0x0008)
[0x0001000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UClass* GFxDataClass; // 0x0080 (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxTourList_TA");
}

return uClassPointer;
};

void HandleTournamentsChanged(class UTourList_TA* InTourList);
void Init(class UGFxShell_X* InShell, class UTourList_TA* InTourList, class UClass* InGFxDataClass);
};

// Class TAGame._InMapDragonManager_TA__IsGoalFirstByTeam_0x1
// 0x0001 (0x0060 - 0x0061)
class U__InMapDragonManager_TA__IsGoalFirstByTeam_0x1 : public UObject

```

```

{
public:
uint8_t           ScoredOnTeam;          // 0x0060 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._InMapDragonManager_TA__IsGoalFirstByTeam_0x1");
}

return uClassPointer;
};

bool __InMapDragonManager_TA__IsGoalFirstByTeam_0x1(class ATeam_TA* T);
};

// Class TAGame.InMapDragonManager_TA
// 0x005C (0x008C - 0x00E8)
class UInMapDragonManager_TA : public UTickableStateObject_X
{
public:
TArray<struct FRotator>      AimOffsets;          // 0x0090 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FRotator>      MaxAims;           // 0x00A0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FRotator>      MinAims;           // 0x00B0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FDragonEntry>   DragonData;         // 0x00C0 (0x0010)
[0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
class AGameEvent_Soccar_TA*   CachedGameEvent;     // 0x00D0
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FName                LastExitAnim;        // 0x00D8 (0x0008)
[0x0000000000002000] (CPF_Transient)
uint8_t                      LastScoredOnTeam;    // 0x00E0 (0x0001)
[0x0000000000002000] (CPF_Transient)
unsigned long                 bViewingReplayFile : 1; // 0x00E4 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InMapDragonManager_TA");
}
}

```

```

return uClassPointer;
};

int32_t __InMapDragonManager_TA__IsGoalFirstByTeam_0x2(int32_t Sum, class ATeam_TA* Element);
void __InMapDragonManager_TA__HandleGoalScored_0x2();
void __InMapDragonManager_TA__HandleGoalScored_0x1();
void SetAllHidden(unsigned long bHidden);
float GetPercentGameTimeRemaining();
bool ShouldExitFly(uint8_t ScoredOnTeam);
bool IsGoalFirstByTeam(uint8_t ScoredOnTeam);
void HandleGameStateChanged(class AGameEvent_TA*_);
void PlayExitAnim();
void HandleGoalScored(class AGameEvent_Soccar_TA* InGameEvent, uint8_t ScoredOnTeam);
void PlayAnimOnComponent(class UAnimStateComponent_TA* Dragon, struct FName AnimName, unsigned long bForce);
void PlayAnimOnAll(struct FName AnimName, unsigned long bForce);
void UnsubscribeFromGameEvent();
void HandleGameEventSet(class AGameEvent_Soccar_TA* InGameEvent);
bool ShouldLerpAim();
void InitDragon(class UAnimStateComponent_TA* InDragon);
void eventTick(float DeltaTime);
void Teardown();
void UnregisterDragon(class UAnimStateComponent_TA* InDragon);
void HandleMapTransition(class FString_);
void RegisterDragon(class UAnimStateComponent_TA* InDragon, int32_t TeamIndex);
};

// Class TAGame.__InMapDragonManager_TA__PlayAnimOnAll_0x1
// 0x000C (0x0060 - 0x006C)
class U__InMapDragonManager_TA__PlayAnimOnAll_0x1 : public UObject
{
public:
    struct FName           AnimName;          // 0x0060 (0x0008)
    [0x0000000000000000]
    unsigned long          bForce : 1;        // 0x0068 (0x0004)
    [0x0000000000000000] [0x00000001]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__InMapDragonManager_TA__PlayAnimOnAll_0x1");
        }
    }

    return uClassPointer;
};

void __InMapDragonManager_TA__PlayAnimOnAll_0x1(struct FDragonEntry D);
};

```

```
// Class TAGame.__InMapDragonManager_TA_SetAllHidden_0x1
// 0x0004 (0x0060 - 0x0064)
class U__InMapDragonManager_TA_SetAllHidden_0x1 : public UObject
{
public:
unsigned long           bHidden : 1;           // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__InMapDragonManager_TA_SetAllHidden_0x1");
}

return uClassPointer;
};

void __InMapDragonManager_TA_SetAllHidden_0x1(struct FDragonEntry D);
};

// Class TAGame.__InMapDragonManager_TA_UnregisterDragon_0x1
// 0x0008 (0x0060 - 0x0068)
class U__InMapDragonManager_TA_UnregisterDragon_0x1 : public UObject
{
public:
class UAnimStateComponent_TA*      InDragon;           // 0x0060 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__InMapDragonManager_TA_UnregisterDragon_0x1");
}

return uClassPointer;
};

bool __InMapDragonManager_TA_UnregisterDragon_0x1(struct FDragonEntry D);
};

// Class TAGame.__Loadout_TA_ApplyCompatibleAttributeProducts_0x1
// 0x0004 (0x0060 - 0x0064)
class U__Loadout_TA_ApplyCompatibleAttributeProducts_0x1 : public UObject
```

```
{  
public:  
int32_t EquippedBodyID; // 0x0060 (0x0004)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._Loadout_TA_ApplyCompatibleAttributeProducts_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __Loadout_TA_ApplyCompatibleAttributeProducts_0x1(class UProduct_TA* B);  
};  
  
// Class TAGame._Loadout_TA_ValidateForcedProducts_0x1  
// 0x000C (0x0060 - 0x006C)  
class U_Loadout_TA_ValidateForcedProducts_0x1 : public UObject  
{  
public:  
class UProductAttribute_TeamForcedProduct_TA* TeamForcedProducts; //  
0x0060 (0x0008) [0x0000000000000000]  
int32_t TeamIndex; // 0x0068 (0x0004)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._Loadout_TA_ValidateForcedProducts_0x1");  
}  
  
return uClassPointer;  
};  
  
int32_t __Loadout_TA_ValidateForcedProducts_0x1(int32_t PID);  
};  
  
// Class TAGame.LoadoutValidation_TA  
// 0x0018 (0x0060 - 0x0078)  
class ULoadoutValidation_TA : public UObject  
{  
public:
```

```
struct FScriptDelegate           _OnLoadoutCorrected_Delegate;      // 0x0060
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutValidation_TA");
}

return uClassPointer;
};

static bool __LoadoutValidation_TA__RemoveIrrelevantAttributes_0x1(class
UProductAttribute_TA* A);
static void SetLoadoutSlotToDefaultID(int32_t TeamIndex, class UProductSlot_TA* Slot, struct
FLoadoutData& TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
static bool ValidateLoadoutTeamPaints(struct FUniqueNetId InPlayerId, struct FLoadoutData&
TeamLoadouts);
static bool ValidateLoadoutSlots(struct FUniqueNetId InPlayerId, struct FLoadoutData&
TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
static void ValidateLoadoutDLC(struct FUniqueNetId InPlayerId, struct FScriptDelegate
OnLoadoutModified, struct FScriptDelegate OnTeamPaintModified, struct FLoadoutData&
TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
static bool ShouldValidateOnlineProducts(struct FUniqueNetId InPlayerId);
static void CorrectDLCOwnership(class APRI_TA* InPRI, struct FScriptDelegate
OnLoadoutModified, struct FScriptDelegate OnTeamPaintModified);
static void RemoveCertifiedProductStat(struct FProductInstanceID InstanceID, TArray<struct
FOnlineProductStat>& OutOnlineProductStats);
static void RemoveIrrelevantAttributes(struct FLoadoutData& Loadout, struct
FLoadoutAttributesArray& OutLoadoutAttributes);
static bool ValidateOnlineProductWithLoadoutProduct(class UOnlineProduct_TA*
LoadoutOnlineProduct, int32_t LoadoutProductId, struct FUniqueNetId InPlayerId);
static bool ValidateTitleIDAttribute(class UProductAttribute_TitleID_TA* Current, class
UProductAttribute_TitleID_TA* Actual);
static bool ValidatePaintedAttribute(class UProductAttribute_Painted_TA* Current, class
UProductAttribute_Painted_TA* Actual);
static bool ValidateCertifiedAttribute(struct FOnlineProductStat ProductStat, class
UProductAttribute_Certified_TA* Actual);
static bool CorrectNonClientAuthoritativeAttributes(class UProductSlot_TA* Slot, struct
FLoadoutAttributesArray& OutAttributes, TArray<class UOnlineProduct_TA*>& PsyNetProducts);
static bool CorrectOnlineData(struct FUniqueNetId InPlayerId, class UProductsConfig_TA*
InProductsConfig, struct FLoadoutData& OutLoadout, struct FLoadoutAttributesArray&
OutAttributes, struct FClientLoadoutOnlineData& OutOnlineLoadout, TArray<struct
FOnlineProductStat>& OutOnlineProductStats, TArray<class UOnlineProduct_TA*>&
OnlineProducts);
static bool CorrectDuplicateAttributes(int32_t TeamIndex, class UProductSlot_TA* Slot, struct
FLoadoutAttributesArray& LoadoutAttributes);
static bool CorrectOfflineData(struct FUniqueNetId InPlayerId, struct FLoadoutData&
TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
void OnLoadoutCorrected();
```

```
};

// Class TAGame.__LoadoutValidation_TA__CorrectDLCOwnership_0x1
// 0x0038 (0x0060 - 0x0098)
class U_LoadoutValidation_TA__CorrectDLCOwnership_0x1 : public UObject
{
public:
    class APRI_TA* InPRI; // 0x0060 (0x0008)
    [0x0000000000000000]
    struct FScriptDelegate OnLoadoutModified; // 0x0068 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate OnTeamPaintModified; // 0x0080 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__LoadoutValidation_TA__CorrectDLCOwnership_0x1");
        }

        return uClassPointer;
    }

void __LoadoutValidation_TA__CorrectDLCOwnership_0x1();
};

// Class TAGame.__MapsConfig_TA__GetMapFrequencyForSet_0x1
// 0x0008 (0x0060 - 0x0068)
class U_MapsConfig_TA__GetMapFrequencyForSet_0x1 : public UObject
{
public:
    struct FName MapName; // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__MapsConfig_TA__GetMapFrequencyForSet_0x1");
        }

        return uClassPointer;
    }

bool __MapsConfig_TA__GetMapFrequencyForSet_0x1(struct FOnlineMapData M);
```

```
};

// Class TAGame.MapsConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UMapsConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FOnlineMapSet>           OnlineMapSets;           // 0x0078 (0x0010)
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapsConfig_TA");
}

return uClassPointer;
};

class UMapData_TA* __MapsConfig_TA__Apply_0x1(struct FOnlineMapData M);
bool GetMapFrequencyForSet(struct FName MapName, struct FName SetName, float& Frequency);
class UMapSet_TA* GetMapSetByName(struct FName SetName);
void Apply();
};

// Class TAGame.__MapsConfig_TA__GetMapSetByName_0x1
// 0x0008 (0x0060 - 0x0068)
class U__MapsConfig_TA__GetMapSetByName_0x1 : public UObject
{
public:
struct FName           SetName;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__MapsConfig_TA__GetMapSetByName_0x1");
}

return uClassPointer;
};

bool __MapsConfig_TA__GetMapSetByName_0x1(struct FOnlineMapSet S);
};
```

```
// Class TAGame.__MapSet_TA__GetMap_0x1
// 0x0008 (0x0060 - 0x0068)
class U__MapSet_TA__GetMap_0x1 : public UObject
{
public:
struct FName MapName; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__MapSet_TA__GetMap_0x1");
}

return uClassPointer;
};

bool __MapSet_TA__GetMap_0x1(class UMapData_TA* M);
};

// Class TAGame.__MatchmakingViewFilter_MenuTree_TA__GetAccessiblePlaylists_0x1
// 0x0008 (0x0060 - 0x0068)
class U__MatchmakingViewFilter_MenuTree_TA__GetAccessiblePlaylists_0x1 : public UObject
{
public:
struct FName LastVisitedPlaylistFolderID; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__MatchmakingViewFilter_MenuTree_TA__GetAccessiblePlaylists_0x1");
}

return uClassPointer;
};

bool __MatchmakingViewFilter_MenuTree_TA__GetAccessiblePlaylists_0x1(class
UMenuTreeNode_TA* N);
};

// Class TAGame.MatchmakingViewFilter_MenuTree_TA
// 0x0018 (0x0060 - 0x0078)
class UMatchmakingViewFilter_MenuTree_TA : public UObject
```

```
{  
public:  
class UIOnlineGamePlaylists_X* [0x0000800000000000] Playlists_Object; // 0x0060 (0x0008)  
class UIOnlineGamePlaylists_X* (0x0008) [0x0000800000000000] Playlists_Interface; // 0x0068  
class UMenuTreeBuilder_TA* (0x0008) [0x0001800000000000] MenuTreeBuilder; // 0x0070  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilter_MenuTree_TA");  
}  
  
return uClassPointer;  
};  
  
bool __MatchmakingViewFilter_MenuTree_TA__GetAccessiblePlaylists_0x3(struct FName PlaylistName);  
struct FName __MatchmakingViewFilter_MenuTree_TA__GetAccessiblePlaylists_0x2(class UMenuTreeNode_Playlist_TA* N);  
TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID, TArray<struct FName>& SelectedPlaylists);  
};  
  
// Class TAGame.__MatchSeries_TA__GetSeriesWinnerFromData_0x1  
// 0x0004 (0x0060 - 0x0064)  
class U__MatchSeries_TA__GetSeriesWinnerFromData_0x1 : public UObject  
{  
public:  
int32_t HalfMatchGames; // 0x0000000000000000  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.__MatchSeries_TA__GetSeriesWinnerFromData_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __MatchSeries_TA__GetSeriesWinnerFromData_0x1(int32_t GamesWon);  
};
```

```

// Class TAGame.MatchSeries_TA
// 0x0030 (0x0060 - 0x0090)
class UMatchSeries_TA : public UObject
{
public:
int32_t SeriesLength; // 0x0060 (0x0004)
[0x0000000000000000]
TArray<struct FSeriesTeam> Teams; // 0x0068 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSeriesGameForfeit__Delegate; // 0x0078
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchSeries_TA");
}

return uClassPointer;
};

int32_t __MatchSeries_TA__GetGamesPlayed_0x1(int32_t Sum, struct FSeriesTeam Team);
int32_t __MatchSeries_TA__GetSeriesWinner_0x1(struct FSeriesTeam T);
static int32_t GetSeriesWinCount(int32_t InSeriesLength);
static int32_t GetSeriesWinnerFromData(int32_t InSeriesLength, TArray<int32_t>& InGamesWon);
int32_t GetSeriesWinner();
int32_t GetGamesPlayed();
bool AllGamesPlayed();
void ForfeitGame(int32_t WinningTeamIndex);
void ScoreGame(int32_t TeamIndex);
int32_t FindTeamForPlayer(struct FUniqueNetId Player);
void RemoveTeamPlayer(struct FUniqueNetId Player);
void AddTeamPlayer(int32_t TeamIndex, struct FUniqueNetId Player);
void EventSeriesGameForfeit(class UMatchSeries_TA* MatchSeries, int32_t WinningTeamIndex);
};

// Class TAGame.__MatchType_Public_TA__CanSkipPreMatchLobby_0x1
// 0x0004 (0x0060 - 0x0064)
class U__MatchType_Public_TA__CanSkipPreMatchLobby_0x1 : public UObject
{
public:
int32_t MaxTeamSize; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__MatchType_Public_TA__CanSkipPreMatchLobby_0x1");
}

return uClassPointer;
};

bool __MatchType_Public_TA__CanSkipPreMatchLobby_0x1(class ATeam_TA* Team);
};

// Class TAGame.MatchType_TA
// 0x003C (0x0060 - 0x009C)
class UMatchType_TA : public UObject
{
public:
class AGameEvent_TA*           GameEvent;          // 0x0060 (0x0008)
[0x0000000000000000]
TArray<struct FComponentTemplate>      Components;        // 0x0068
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long                  bPreAssignedTeams : 1; // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
int32_t                         ForfeitVoteWaitGameTime; // 0x007C (0x0004)
[0x0000004000000000]
int32_t                         WaitForPlayersTime; // 0x0080 (0x0004)
[0x0000000000000000]
float                           WaitingForPlayersStartTime; // 0x0084 (0x0004)
[0x0000000000000000]
struct FName                    MatchTypeName; // 0x0088 (0x0008)
[0x0000000000000000]
float                           LobbyTime;          // 0x0090 (0x0004)
[0x0000000000000000]
float                           LobbyIntroTime; // 0x0094 (0x0004)
[0x0000000000000002] (CPF_Const)
int32_t                         LobbyEndCountdown; // 0x0098 (0x0004)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_TA");
}

return uClassPointer;
};

struct FUniqueNetId __MatchType_TA__AllPlayersHaveACar_0x2(class AController* P);
bool __MatchType_TA__AllPlayersHaveACar_0x1(class AController* P);
class ATeam_TA* PickTeamFromReservations(class AController* C);

```

```

void OnInitGameEvent();
bool ShouldKickOnMigrate();
bool ChooseTeam(int32_t TeamIndex, class APlayerController_TA* Player);
bool CanRestartPlayer(class AController* NewPlayer);
bool ShouldAutoSelectTeam(class AController* NewPlayer);
bool AutoSpectate(class AController* NewPlayer);
bool ShouldShowWaitingForPlayersTime();
bool ShouldSubmitMatchComplete();
bool CanCheckForBannedPlayers();
bool CanCheckForForfeitSeries();
bool UseCustomMatchSettings();
bool CanIncreaseMaxTeamSize();
bool IsOnlineMultiplayer();
bool ShouldSetGameOwner();
bool CanCheckForForfeit();
bool ShouldCancelMatch();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
bool AllowMatchAdmin();
bool RecordPlayedMap();
bool AllowSpectators();
bool CanStayAsParty();
bool CanRematchVote();
bool CanUpdateStats();
bool GiveExtraXP();
struct FName GetMatchTypeName();
void OnStartWaitingForPlayers();
void OnPlayerAddedToTeam();
bool ShouldShowDisconnectedPlayersOnScoreboard();
bool CanSkipPreMatchLobby();
bool ShowPreMatchLobby();
class ATeam_TA* PickTeam(class AController* C);
int32_t GetMaxTeamSize();
int32_t GetWaitTimeRemaining();
class UOnlineGameDedicatedServer_TA* GetOnlineGameDedicatedServer();
class UOnlineGame_Base_X* GetOnlineGame();
bool PlayerNeedsCar(class AController* Player);
bool AllPlayersHaveACar(TArray<struct FUniqueNetId>& OutPlayersWithACar);
void InitCustomMatchSettings(struct FCustomMatchSettings Settings);
void Init(class AGameEvent_TA* InGameEvent, class FString Options);
};

// Class TAGame.MatchType_Public_TA
// 0x000C (0x009C - 0x00A8)
class UMatchType_Public_TA : public UMatchType_TA
{
public:
    class UOnlineGameReservations_X*             Reservations;           // 0x00A0
    (0x0008) [0x0000800000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Public_TA");
}

return uClassPointer;
};

bool __MatchType_Public_TA__TeamsHaveEnoughHumans_0x1(class ATeam_TA* Team);
bool CanRestartPlayer(class AController* NewPlayer);
bool ShouldAutoSelectTeam(class AController* NewPlayer);
bool CanCheckForBannedPlayers();
bool ShouldSubmitMatchComplete();
bool IsOnlineMultiplayer();
bool CanUpdateStats();
bool CanStayAsParty();
bool CanRematchVote();
bool CanSkipPreMatchLobby();
class ATeam_TA* PickTeamFromReservations(class AController* C);
bool IsWaitingForPlayersTimeExpired();
bool CanCheckForForfeit();
bool TeamsHaveEnoughHumans();
bool ShouldCancelMatch();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
void OnStartWaitingForPlayers();
bool GiveExtraXP();
};

// Class TAGame.__MatchType_Tournament_TA__CanSkipPreMatchLobby_0x2
// 0x0004 (0x0060 - 0x0064)
class U__MatchType_Tournament_TA__CanSkipPreMatchLobby_0x2 : public UObject
{
public:
int32_t MaxTeamSize; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__MatchType_Tournament_TA__CanSkipPreMatchLobby_0x2");
}

return uClassPointer;
};

bool __MatchType_Tournament_TA__CanSkipPreMatchLobby_0x2(class ATeam_TA* Team);
};

// Class TAGame.MatchType_PublicRanked_TA

```

```

// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_PublicRanked_TA : public UMatchType_Public_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_PublicRanked_TA");
}

return uClassPointer;
};

bool ShouldCancelMatch();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
};

// Class TAGame.MatchType_Tournament_TA
// 0x0028 (0x00A8 - 0x00D0)
class UMatchType_Tournament_TA : public UMatchType_PublicRanked_TA
{
public:
class UOnlineGameDedicatedServer_TA* DedicatedServer; // 0x00A8
(0x0008) [0x0001800000000001] (CPF_Edit)
class UOnlineGameTourServer_TA* TourServer; // 0x00B0
(0x0008) [0x0001800000000001] (CPF_Edit)
class UTourServerConfig_TA* TournamentServerConfig; // 0x00B8
(0x0008) [0x0001800000000001] (CPF_Edit)
class UMatchSeries_TA* MatchSeries; // 0x00C0 (0x0008)
[0x0001800000000001] (CPF_Edit)
struct FName PlaylistName; // 0x00C8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Tournament_TA");
}

return uClassPointer;
};

struct FUniqueNetId __MatchType_Tournament_TA__ShouldStartMatch_0x1(class APRI_TA* PRI);
bool __MatchType_Tournament_TA__CanSkipPreMatchLobby_0x1(class ATeam_TA* T);

```

```

bool CanUpdateStats();
bool CanCheckForBannedPlayers();
bool UseCustomMatchSettings();
bool ShouldSubmitMatchComplete();
bool CanCancelMatch();
bool CanRematchVote();
bool ShouldKickOnMigrate();
bool CanSkipPreMatchLobby();
bool ShowPreMatchLobby();
bool IsFirstGameInMatch();
bool IsBotOnlyTeam(class ATeam_TA* Team);
bool HasBotTeams();
bool ShouldShowWaitingForPlayersTime();
bool CanCheckForForfeitSeries();
bool CanCheckForForfeit();
bool ShouldCancelMatch();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
int32_t GetMaxTeamSize();
bool PlayerHasTeam(class AController* Player);
bool PlayerNeedsCar(class AController* Player);
bool AutoSpectate(class AController* NewPlayer);
bool CanRestartPlayer(class AController* NewPlayer);
class ATeam_TA* PickTeam(class AController* C);
bool ShouldAutoSelectTeam(class AController* NewPlayer);
void OnStartWaitingForPlayers();
void OnInitGameEvent();
};

// Class TAGame.__MenuSequencer_TA__GetSequence_0x1
// 0x0008 (0x0060 - 0x0068)
class U__MenuSequencer_TA__GetSequence_0x1 : public UObject
{
public:
    struct FName SequenceName; // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__MenuSequencer_TA__GetSequence_0x1");
        }

        return uClassPointer;
    };

    bool __MenuSequencer_TA__GetSequence_0x1(class UMenuSequence_TA* S);
};

// Class TAGame.MenuSequence_TA
// 0x0048 (0x0060 - 0x00A8)

```

```

class UMenuSequence_TA : public UObject
{
public:
struct FName CameraState; // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long bRemotePlayersHidden : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bLockPreviewActor : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bImmediate : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
class UMenuSequencer_TA* MenuSequenceComp; // 0x0070
(0x0008) [0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FScriptDelegate __EventEnteredSequence__Delegate; // 0x0078
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventLeftSequence__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuSequence_TA");
}

return uClassPointer;
};

class ACamera_MainMenu_TA* GetCamera();
void SetLockPreviewActor();
void SetRemotePlayersHidden();
void SetCameraState();
void TickSequence(float DeltaTime);
void LeaveSequence(class UMenuSequence_TA* NextSequence);
void EnterSequence(class UMenuSequence_TA* PrevSequence);
void Init(class UMenuSequencer_TA* InComp);
void EventLeftSequence(class UMenuSequence_TA* InSequence);
void EventEnteredSequence(class UMenuSequence_TA* InSequence);
};

// Class TAGame.MenuSequencer_TA
// 0x005C (0x00A4 - 0x0100)
class UMenuSequencer_TA : public UActorComponent_X
{
public:
TArray<class UMenuSequence_TA*> Sequences; // 0x00A8
(0x0010) [0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FName DefaultSequence; // 0x00B8 (0x0008)
[0x0000000000000001] (CPF>Edit)
class UMenuSequence_TA* CurrentSequence; // 0x00C0
(0x0008) [0x0000004000000000]

```

```

class UMenuSequence_TA*           PendingSequence;          // 0x00C8
(0x0008) [0x0000000000000000]
uint8_t                          Status;                // 0x00D0 (0x0001)
[0x0000000000000000]
TArray<class UMenuSequence_TA*> PendingSequenceQueue;    //
0x00D8 (0x0010) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventSequenceChanged__Delegate; // 0x00E8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuSequencer_TA");
}

return uClassPointer;
};

class APlayerController* GetLocalPlayerController();
bool IsSequencePending(struct FName SequenceName);
bool IsSequenceActive(struct FName SequenceName);
void eventTick(float DeltaTime);
void HandleSequenceEntered(class UMenuSequence_TA* Sequence);
void InternalSequenceEnter(class UMenuSequence_TA* NewSequence, class
UMenuSequence_TA* PrevSequence);
void InternalSequenceLeave(class UMenuSequence_TA* NewSequence, class
UMenuSequence_TA* PrevSequence);
void HandleSequenceLeft(class UMenuSequence_TA* FromSequence);
void TryEnterNextSequence();
void SetSequence(struct FName SequenceName);
class UMenuSequence_TA* GetSequence(struct FName SequenceName);
class UMenuSequence_TA* InstanceSequence(class UMenuSequence_TA* Archetype);
void InstanceSequences();
void eventAttached();
void EventSequenceChanged(class UMenuSequence_TA* NewSequence);
};

// Class TAGame.__MenuTreeBranch_TA__OnChildNodeClicked_0x1
// 0x0008 (0x0060 - 0x0068)
class U__MenuTreeBranch_TA__OnChildNodeClicked_0x1 : public UObject
{
public:
struct FName                  ChildID;                // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__MenuTreeBranch_TA__OnChildNodeClicked_0x1");
}

return uClassPointer;
};

bool __MenuTreeBranch_TA__OnChildNodeClicked_0x1(class UMenuTreeNode_TA* Child);
};

// Class TAGame.MenuTreeBranch_TA
// 0x0038 (0x0178 - 0x01B0)
class UMenuTreeBranch_TA : public UMenuTreeNode_TA
{
public:
TArray<struct FName> ChildIDs; // 0x0178 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<class UMenuTreeNode_TA*> ChildNodes; // 0x0188
(0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __EventBranchNodeClicked__Delegate; // 0x0198
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeBranch_TA");
}

return uClassPointer;
};

bool __MenuTreeBranch_TA__GetActiveNodes_0x1(class UMenuTreeNode_TA* N);
class UMenuTreeBranch_TA* SetChildren(TArray<class UMenuTreeNode_TA*> InChildren);
TArray<class UMenuTreeNode_TA*> GetActiveNodes();
void OnChildNodeClicked(struct FName ChildID);
void OnClicked();
void EventBranchNodeClicked(class UMenuTreeBranch_TA* ClickedNode);
};

// Class TAGame.__MenuTreeBuilder_TA__BuildTree_0x1
// 0x0008 (0x0060 - 0x0068)
class U__MenuTreeBuilder_TA__BuildTree_0x1 : public UObject
{
public:
struct FName ChildID; // 0x0060 (0x0008)
[0x0001000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__MenuTreeBuilder_TA__BuildTree_0x1");
}

return uClassPointer;
};

bool __MenuTreeBuilder_TA__BuildTree_0x1(class UMenuTreeNode_TA* N);
};

// Class TAGame.MenuTreeBuilder_TA
// 0x0020 (0x0060 - 0x0080)
class UMenuTreeBuilder_TA : public UObject
{
public:
class UMenuTreePlaylistManager_TA*           PlaylistManager;           // 0x0060
(0x0008) [0x0001000000000000]
TArray<class UMenuTreeNode_TA*>           AllNodes;                 // 0x0068
(0x0010) [0x0001004004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UObjectReferencer*                   PlaylistIconReferences; // 0x0078 (0x0008)
[0x0001000000000003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeBuilder_TA");
}

return uClassPointer;
};

void __MenuTreeBuilder_TA__Initialize_0x1(class UMenuTreeNode_TA* N);
int32_t __MenuTreeBuilder_TA__SortNodes_0x1(class UMenuTreeNode_TA* A, class
UMenuTreeNode_TA* B);
void ValidateCircularity();
void ValidateUniqueness();
void BuildTree();
void SortNodes();
void Initialize();
};

// Class TAGame.RPC_GetActivePlaylists_TA
// 0x0024 (0x00E8 - 0x010C)
class URPC_GetActivePlaylists_TA : public URPC_X

```

```

{
public:
TArray<struct FActivePlaylistInfo>           CasualPlaylists;          // 0x00E8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FActivePlaylistInfo>           RankedPlaylists;        // 0x00F8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                           XPLevelUnlocked;      // 0x0108 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetActivePlaylists_TA");
}

return uClassPointer;
};

void eventOnSuccess();
};

// Class TAGame.__MenuTreePlaylistManager_TA__HandlePlaylistRPCComplete_0x1
// 0x0010 (0x0060 - 0x0070)
class U__MenuTreePlaylistManager_TA__HandlePlaylistRPCComplete_0x1 : public UObject
{
public:
uint64_t           CurrentTime;          // 0x0060 (0x0008)
[0x0001000000000000]
class UMenuTreeNode_Playlist_TA*       PlaylistNode;        // 0x0068
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__MenuTreePlaylistManager_TA__HandlePlaylistRPCComplete_0x1");
}

return uClassPointer;
};

bool __MenuTreePlaylistManager_TA__HandlePlaylistRPCComplete_0x2(struct
FActivePlaylistInfo AP);
bool __MenuTreePlaylistManager_TA__HandlePlaylistRPCComplete_0x1(struct
FActivePlaylistInfo P);
};

```

```

// Class TAGame.MenuTreeNode_Playlist_TA
// 0x006058 (0x0178 - 0x01D80)
class UMenuTreeNode_Playlist_TA : public UMenuTreeNode_TA
{
public:
int32_t PlaylistId; // 0x0178 (0x0004)
[0x0001000000000000]
int32_t PlaylistRowIndex; // 0x017C (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
class UGFxData_Playlist_TA* Playlist; // 0x0180 (0x0008)
[0x0001000000000000]
int32_t MinCompetitiveXPLevelRequired; // 0x0188 (0x0004)
[0x0001000000000000]
class UMenuTreePlaylistQueue_TA* PlaylistQueue; // 0x0190
(0x0008) [0x0001000000000000]
unsigned long bSelected : 1; // 0x0198 (0x0004)
[0x0001004000000000] [0x00000001]
uint8_t SpecialDisplayType; // 0x019C (0x0001)
[0x0001000040000000] (CPF_EditInlineNotify)
class UMenuTreeConfig_TA* MenuTreeConfig; // 0x01A0
(0x0008) [0x0001800000000000]

struct FScriptDelegate __EventPlaylistNodeClicked__Delegate; // 0x01A80
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPlaylistNodeUpdated__Delegate; // 0x01C0B8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_Playlist_TA");
}

return uClassPointer;
};

bool __MenuTreeNode_Playlist_TA__HandleFirstTimeClicked_0x1(int32_t PID);
bool __MenuTreeNode_Playlist_TA__HandleLockForGatedPlaylists_0x2(struct FPartyMember InPartyMember);
bool __MenuTreeNode_Playlist_TA__HandleLockForGatedPlaylists_0x1(class ULocalPlayer_TA* InLocalPlayer);
bool HandleLockForGatedPlaylists();
void UpdateLock();
void UpdateSelected(unsigned long bNewSelected);
void UpdateBadge();
void UpdateSecondaryTitle();
void UpdatePlaylistInfo();
class FString GetPlaylistName(class UGFxShell_X* InShell);
uint64_t GetEndTime();
uint8_t GetActivePlaylistType();

```

```

void HandleActivePlaylistUpdated(struct FActivePlaylistInfo NewActivePlaylist);
void SetLastMatchmakePlaylistID(class FString InID);
void HandlePrivilegeCheck(class UPrivilegeCheck_X* PrivilegeCheck);
bool HandleFirstTimeClicked();
void OnClicked();
void OnGfxNodeSet();
void OnInit();
void EventPlaylistNodeUpdated(class UMenuTreeNode_Playlist_TA* PlaylistNode);
void EventPlaylistNodeClicked(class UMenuTreeNode_Playlist_TA* PlaylistNode);
};

// Class TAGame.MenuTreePlaylistManager_TA
// 0x0038 (0x0060 - 0x0098)
class UMenuTreePlaylistManager_TA : public UObject
{
public:
    class UMenuTreeBuilder_TA*           MenuTreeBuilder;          // 0x0060
    (0x0008) [0x0001000000000000]
    TArray<struct FActivePlaylistInfo>   ActiveAndFuturePlaylists; // 0x0068
    (0x0010) [0x000100000400000] (CPF_NeedCtorLink)
    class UMenuTreeConfig_TA*            MenuTreeConfig;          // 0x0078
    (0x0008) [0x0001800000000000]
    struct FScriptDelegate              __EventActivePlaylistsUpdated__Delegate; // 0x0080
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.MenuTreePlaylistManager_TA");
        }

        return uClassPointer;
    }

void HandlePlaylistRPCComplete(class URPC_GetActivePlaylists_TA* InRPC);
void ClearRequestTimer();
void RequestPlaylists();
void Init(class UMenuTreeBuilder_TA* InMenuTreeBuilder);
void EventActivePlaylistsUpdated(class UMenuTreePlaylistManager_TA* InManager);
};

// Class TAGame._MtxConfig_TA__CreateModalForError_0x1
// 0x0008 (0x0060 - 0x0068)
class U_MtxConfig_TA__CreateModalForError_0x1 : public UObject
{
public:
    class UError*                      Error;                  // 0x0060 (0x0008)
    [0x0000000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._MtxConfig_TA__CreateModalForError_0x1");
}

return uClassPointer;
};

void __MtxConfig_TA__CreateModalForError_0x1(class UGFxModal_X* Modal);
};

// Class TAGame.__Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x1
// 0x0008 (0x0060 - 0x0068)
class U__Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x1 : public UObject
{
public:
class UGameplaySettingsSave_TA*           Settings;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x1");
}

return uClassPointer;
};

void __Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x4();
void __Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x3();
void __Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x2();
void __Mutator_Freeplay_TA__HandlePrimaryGameplaySettingsSave_0x1();
};

// Class TAGame.Mutator_Freeplay_TA
// 0x004C (0x0064 - 0x00B0)
class UMutator_Freeplay_TA : public UMutator_TA
{
public:
class ACar_TA*           CarArchetype;           // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<class ATeam_Soccar_TA*>     TeamArchetypes;     // 0x0070
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class AGameEvent_Soccar_TA*      SoccarGame;      // 0x0080

```

```

(0x0008) [0x0000000000000000] TArray<struct FRumbleItemIndex> PlayerRumbleItems; // 0x0088
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
float LastResetTime; // 0x0098 (0x0004)
[0x0000000000000000]
float ResetCooldown; // 0x009C (0x0004)
[0x0000000000000002] (CPF_Const)
class UFreeplayConfig_TA* FreeplayConfig; // 0x00A0 (0x0008)
[0x0008000000000000]
uint8_t BoostFillType; // 0x00A8 (0x0001)
[0x0000000000000000]
float BoostFillDelay; // 0x00AC (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Freeplay_TA");
}

return uClassPointer;
};

void __Mutator_Freeplay_TA__Init_0x3(class UGameplaySettingsSave_TA* Settings);
void __Mutator_Freeplay_TA__Init_0x2(class UGoal_TA* G);
void __Mutator_Freeplay_TA__Init_0x1(class AGameEvent_TA* SG, class ACar_TA* InCar);
TArray<class AActor*> __Mutator_Freeplay_TA__HandleTeams_0x1(class ATeam_Soccar_TA* T);
void __Mutator_Freeplay_TA__SetBoostFillForCar_0x1(class ACarComponent_Boost_TA* Boost);
static void HandleNewBoostFillSetting(class ACarComponent_Boost_TA* InBoost, uint8_t InBoostFillType, float InBoostFillDelay, class UFreeplayConfig_TA* InFreeplayConfig);
void SetBoostFillForCar(class ACar_TA* InCar);
void HandleBoostFillDelayChange(float InBoostFillDelay);
void HandleBoostFillTypeChange(uint8_t InBoostFillType);
void HandleGoalDisableChange(unsigned long bDisableGoals);
void HandleGameSpeedChange(float NewSpeed);
void HandlePrimaryGameplaySettingsSave(class UGameplaySettingsSave_TA* Settings);
class ASpecialPickup_TA* GetNextRumbleItem(class UGameEvent_Soccar_SubRules_Items_TA* Items, class APlayerController_TA* PC);
void GiveNextRumbleItem(class ACar_TA* Car);
void HandleAttachedPickup(class ACar_TA* Car);
void MutateObject(class UObject* O);
void HandleTeams(class AGameEvent_Team_TA*_);
void HandlePlayerResetTraining(class AGameEvent_TA*_);
void HandleLocalPlayerShell(class UGFxShell_X* Shell);
void HandleLocalPlayer(class ULocalPlayer_TA* P);
void SpawnDummyBot();
bool ShouldStartRound();
void CheckStartRound();
void HandleGameStateChanged(class AGameEvent_TA*_);
void Init(class AGameEvent_TA* GameEvent);

```

```
};

// Class TAGame.__Mutator_Freeplay_TA__MutateObject_0x1
// 0x0008 (0x0060 - 0x0068)
class U_Mutator_Freeplay_TA__MutateObject_0x1 : public UObject
{
public:
class ACar_TA*           Car;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Mutator_Freeplay_TA__MutateObject_0x1");
}

return uClassPointer;
};

void __Mutator_Freeplay_TA__MutateObject_0x1();
};

// Class TAGame.__Mutator_Robin_TA__Construct_0x1
// 0x0008 (0x0060 - 0x0068)
class U_Mutator_Robin_TA__Construct_0x1 : public UObject
{
public:
class ACarComponent_TA*   Override;    // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Mutator_Robin_TA__Construct_0x1");
}

return uClassPointer;
};

bool __Mutator_Robin_TA__Construct_0x1(class ACarComponent_TA* Original);
};

// Class TAGame.Mutator_Robin_TA
// 0x0028 (0x0060 - 0x0088)
class UMutator_Robin_TA : public UObject
```

```
{  
public:  
struct FStickyForceData           StickyForce;          // 0x0060 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
struct FAutoFlipData             AutoFlip;            // 0x0068 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
class UVehicleSim_TA*           VehicleSim;          // 0x0070 (0x0008)  
[0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)  
TArray<class ACarComponent_TA*>   OverrideCarComponents; // 0x0078  
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Robin_TA");  
}  
  
return uClassPointer;  
};  
  
void eventConstruct();  
};  
  
// Class TAGame.__OnlineDLCProductCache_TA__GetOldGeneratedOnlineProduct_0x1  
// 0x0010 (0x0060 - 0x0070)  
class U__OnlineDLCProductCache_TA__GetOldGeneratedOnlineProduct_0x1 : public UObject  
{  
public:  
struct FProductInstanceID         InstanceID;          // 0x0060 (0x0010)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__OnlineDLCProductCache_TA__GetOldGeneratedOnlineProduct_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __OnlineDLCProductCache_TA__GetOldGeneratedOnlineProduct_0x1(class  
UOnlineProduct_TA* P);  
};  
  
// Class TAGame.OnlineDLCProductCache_TA
```

```

// 0x0048 (0x0060 - 0x00A8)
class UOnlineDLCProductCache_TA : public UObject
{
public:
TArray<class UOnlineProduct_TA*> GeneratedProducts; // 0x0060
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
TArray<class UClass*> AllowedAttributes; // 0x0070 (0x0010)
[0x0000004000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<class UOnlineProduct_TA*> PreviousGeneratedProducts; // 0x0080
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __EventDLCChange__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineDLCProductCache_TA");
}

return uClassPointer;
};

class UOnlineProduct_TA* GetOldGeneratedOnlineProduct(struct FProductInstanceID
InstanceID);
void ClearProducts();
void RemoveGeneratedProduct(int32_t ProductID);
void RemoveDuplicateGeneratedItems();
TArray<class FString> GetCurrentFakeEntitlements();
bool Contains(int32_t ProductID);
class UOnlineProduct_TA* GenerateOnlineDLCProduct(int32_t ProductID);
void GetDLCProducts(TArray<class UProduct_TA*>& Unlocked, TArray<class UProduct_TA*>&
Locked);
void HandleUnlockedDLCChanged();
void RegenerateOnlineDLCProducts();
void EventDLCChange();
};

// Class TAGame.__OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheError_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheError_0x1 : public UObject
{
public:
TArray<struct FUniqueNetId> RequestedPlayers; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheError_0x1");
}

return uClassPointer;
};

void __OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheError_0x1(struct
FVoiceRoomTokenRequest P);
};

// Class TAGame.OnlineGameDedicatedServer_TA
// 0x0058 (0x03E0 - 0x0438)
class UOnlineGameDedicatedServer_TA : public UOnlineGameDedicatedServer_X
{
public:
class UOnlineGameTourServer_TA*          Tournaments;           // 0x03E0
(0x0008) [0x0001000000000001] (CPF_Edit)
int32_t                               LastPlaylistID;        // 0x03E8 (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<class UObject*>                AllRelevantProducts; // 0x03F0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UReplay_TA*                      ReplayToUpload;       // 0x0400 (0x0008)
[0x0000000000000000]
class ARLBot_SessionRecorder_TA*       FlatbufferSession;    // 0x0408
(0x0008) [0x0001000000000000]
class UEOSVoiceTokenCache_TA*          VoiceTokenCache;     // 0x0410
(0x0008) [0x0001000000000000]
class UGameModesConfig_TA*             GameModesConfig;    // 0x0418
(0x0008) [0x0000800000000000]
struct FScriptDelegate               __EventGameEventInitialized__Delegate; // 0x0420
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameDedicatedServer_TA");
}

return uClassPointer;
};

void HandleActiveRoundChanged(class AGameEvent_Soccar_TA* GameEvent);
int32_t GetTeamScore(int32_t TeamIdx);
void HandleVoiceTokenCacheError(TArray<struct FVoiceRoomTokenRequest> Requests, class
UError* RequestError);
void HandleVoiceTokenCacheResponse(TArray<struct FVoiceRoomTokenRequest> Requests,
TArray<struct FPlayerVoiceRoomCredentials> NewCredentials);

```

```

void RequestVoiceRoomToken(struct FUniqueNetId PlayerID, class FString RoomId, unsigned long bClearCache);
bool DidRematchVotePass();
void UploadFlatbuffersSession(class ARLBot_SessionRecorder_TA* Session, class FString URL, TArray<struct FKeyValuePair>& Headers);
void InitFlatbufferUpload(class URPC_GetUploadUrls_TA* RPC, class UMatchData_X* Match, class ARLBot_SessionRecorder_TA* Session);
void InitReplayUpload(class URPC_GetUploadUrls_TA* RPC, class UReplay_TA* Replay);
void InitMatchLogUpload(class URPC_GetUploadUrls_TA* RPC, class UMatchLog_X* Log);
void UploadMatchFiles();
bool AllowServerMigration();
bool CanStartMatch();
bool AllowSplitscreenJoinRankedMatch();
bool AllowSplitscreenJoin(struct FUniqueNetId PrimaryPlayerId, struct FUniqueNetId PlayerID, class FString PlayerName, class FString& Error);
void AllowPlayerLogin(class FString Options, struct FUniqueNetId PlayerID, class FString& ErrorMessage);
void GoToNextMap();
static class FString JoinOption(class FString FullString, class FString Value);
struct FName GetNextRandomMapName(int32_t GameMode, unsigned long bStandardOnly, unsigned long bRocketLabsMedleyOnly);
bool IsGameModeEnabled(int32_t GameMode);
void GotoPrivateMatchMap(struct FName MapName, int32_t GameMode, class FString GameTags);
uint8_t GetTeamNum(struct FUniqueNetId PlayerID);
void ReportLowFPS();
int32_t GetPlaylistPlayerCount(class UGameSettingPlaylist_X* Playlist);
void SetPlaylist(int32_t PlaylistId);
void OnMainEventDestroyed();
class FString GetKickReason();
bool KickPlayersOnGameEventDestroyed();
void HandleMigrationJoin();
void ClearSettings();
void HandleGameEventEnded(class AGameEvent_Soccar_TA* GameEvent);
void HandleGameInitialized(class AGameInfo_TA* Game);
void StartMatch();
class AGameEvent_TA* GetCurrentGameEvent();
void LoadAllRelevantProducts();
void HandleRLBotMatchEnded(class AGameEvent_Soccar_TA* _);
void HandleRLBotMatchStarted(class AGameEvent_Soccar_TA* _);
void HandleReplayDirector(class AGameEvent_Soccar_TA* G);
void OnInit();
void EventGameEventInitialized(class AGameEvent_TA* GameEvent);
};

// Class TAGame.___OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheResponse_0x1
// 0x0058 (0x0060 - 0x00B8)
class U___OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheResponse_0x1 : public UObject
{
public:
TArray<struct FUniqueNetId> RequestedPlayers; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FUniqueNetId PlayerID; // 0x0070 (0x0048)
}

```

[0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheResponse_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheResponse_0x2(struct  
FPlayerVoiceRoomCredentials P);  
void __OnlineGameDedicatedServer_TA__HandleVoiceTokenCacheResponse_0x1(struct  
FVoiceRoomTokenRequest P);  
};  
  
// Class TAGame.__OnlineGameDedicatedServer_TA__InitFlatbufferUpload_0x1  
// 0x0010 (0x0060 - 0x0070)  
class U__OnlineGameDedicatedServer_TA__InitFlatbufferUpload_0x1 : public UObject  
{  
public:  
class URPC_GetUploadUrls_TA*           RPC;          // 0x0060 (0x0008)  
[0x0000000000000000]  
class ARLBot_SessionRecorder_TA*       Session;      // 0x0068 (0x0008)  
[0x0001000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__OnlineGameDedicatedServer_TA__InitFlatbufferUpload_0x1");  
}  
  
return uClassPointer;  
};  
  
void __OnlineGameDedicatedServer_TA__InitFlatbufferUpload_0x1(class URPC_X* _);  
};  
  
// Class TAGame.RPC_GetUploadUrls_TA  
// 0x0078 (0x00E8 - 0x0160)  
class URPC_GetUploadUrls_TA : public URPC_X  
{  
public:
```

```

class UGetUploadUrlsMatch_TA*           Match;          // 0x00E8 (0x0008)
[0x0000000000000000]
class FString                         ReplayContentType; // 0x00F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                         LogContentType;   // 0x0100 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                         FlatbufferContentType; // 0x0110 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                         UploadReplayUrl; // 0x0120 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                         UploadLogFile;    // 0x0130 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                         UploadFlatbufferUrl; // 0x0140 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FKeyValuePair>          FlatBufferMetaData; // 0x0150
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetUploadUrls_TA");
}

return uClassPointer;
};

class URPC_GetUploadUrls_TA* SetSaveFlatbuffer(class FString ContentType);
class URPC_GetUploadUrls_TA* SetSaveLog(class FString ContentType);
class URPC_GetUploadUrls_TA* SetSaveReplay(class FString ContentType);
class URPC_GetUploadUrls_TA* SetMatch(class UMatchData_X* InMatch);
};

// Class TAGame.__OnlineGameDedicatedServer_TA__InitMatchLogUpload_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameDedicatedServer_TA__InitMatchLogUpload_0x1 : public UObject
{
public:
class UMatchLog_X*                  Log;            // 0x0060 (0x0008)
[0x0000000000000000]
class URPC_GetUploadUrls_TA*        RPC;           // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class"

```

```
TAGame.__OnlineGameDedicatedServer_TA__InitMatchLogUpload_0x1");
}

return uClassPointer;
};

void __OnlineGameDedicatedServer_TA__InitMatchLogUpload_0x1(class URPC_X* _);
};

// Class TAGame.__OnlineGameDedicatedServer_TA__InitReplayUpload_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameDedicatedServer_TA__InitReplayUpload_0x1 : public UObject
{
public:
class URPC_GetUploadUrls_TA*           RPC;          // 0x0060 (0x0008)
[0x0000000000000000]
class UBinaryUploader_TA*              Uploader;    // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameDedicatedServer_TA__InitReplayUpload_0x1");
}

return uClassPointer;
};

void __OnlineGameDedicatedServer_TA__InitReplayUpload_0x1(class URPC_X* _);
};

// Class TAGame.__OnlineGameDLC_TA__GetDlcIdFromName_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OnlineGameDLC_TA__GetDlcIdFromName_0x1 : public UObject
{
public:
struct FName                         DLCName;     // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameDLC_TA__GetDlcIdFromName_0x1");
}
}
```

```

return uClassPointer;
};

bool __OnlineGameDLC_TA__GetDlcIdFromName_0x1(class UDLCPack_TA* D);
};

// Class TAGame.OnlineGameDLC_TA
// 0x0000 (0x00F0 - 0x00F0)
class UOnlineGameDLC_TA : public UOnlineGameDLC_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameDLC_TA");
}

return uClassPointer;
};

static class FString GetDlcIdFromName(struct FName DLCName, uint8_t PlatformType);
};

// Class TAGame.__OnlineGameParty_TA__AddPlayerPartyServiceLobbyMessage_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameParty_TA__AddPlayerPartyServiceLobbyMessage_0x1 : public UObject
{
public:
class FString                               Message;                                // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameParty_TA__AddPlayerPartyServiceLobbyMessage_0x1");
}

return uClassPointer;
};

void __OnlineGameParty_TA__AddPlayerPartyServiceLobbyMessage_0x1(class UPersona_TA* P);
};

```

```
// Class TAGame.__OnlineGameParty_TA__GetPartyMessageError_0x3
// 0x0008 (0x0060 - 0x0068)
class U__OnlineGameParty_TA__GetPartyMessageError_0x3 : public UObject
{
public:
class UOnlinePlayer_TA*           OnlinePlayer;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameParty_TA__GetPartyMessageError_0x3");
}

return uClassPointer;
};

bool __OnlineGameParty_TA__GetPartyMessageError_0x3(struct FUniqueNetId P);

// Class TAGame.__OnlineGameParty_TA__GetProfileProducts_0x1
// 0x0014 (0x0060 - 0x0074)
class U__OnlineGameParty_TA__GetProfileProducts_0x1 : public UObject
{
public:
TArray<struct FOnlineProductData>      ProfileProductData;    // 0x0060
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)
int32_t                      SlotIndex;                // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameParty_TA__GetProfileProducts_0x1");
}

return uClassPointer;
};

void __OnlineGameParty_TA__GetProfileProducts_0x1(class UProductAttribute_TA* P);

// Class TAGame.__OnlineGameParty_TA__HandleCrossplayEnabledChanged_0x1
```

```

// 0x0008 (0x0060 - 0x0068)
class U__OnlineGameParty_TA__HandleCrossplayEnabledChanged_0x1 : public UObject
{
public:
class UAccountSettingsComponent_TA*      AccountSettings;          // 0x0060
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameParty_TA__HandleCrossplayEnabledChanged_0x1");
}

return uClassPointer;
};

void __OnlineGameParty_TA__HandleCrossplayEnabledChanged_0x1();
};

// Class TAGame.AccountSettingsComponent_TA
// 0x0060 (0x0070 - 0x00D0)
class UAccountSettingsComponent_TA : public UComponent
{
public:
unsigned long              bCrossPlatformEnabled : 1;           // 0x0070 (0x0004)
[0x0008000000000000] [0x00000001]
unsigned long              bPlatformAllowsCrossPlatform : 1;    // 0x0070
(0x0004) [0x0000008000000000] [0x00000002]
unsigned long              bPlatformAllowsInvites : 1;        // 0x0070 (0x0004)
[0x0000008000000000] [0x00000004]
uint8_t                   CrossPlatformChatState;           // 0x0074 (0x0001)
[0x0008008000000000]
class UNetworkSave_TA*      NetworkSave;                  // 0x0078 (0x0008)
[0x0000000000000000]
class UPlatformAccountSettings* PlatformSettings;          // 0x0080
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FScriptDelegate      __EventAccountSettingsChanged__Delegate; // 0x0088
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __bCrossPlatformEnabled__ChangeNotify; // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __CrossPlatformChatState__ChangeNotify; // 0x00B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.AccountSettingsComponent_TA");
}

return uClassPointer;
};

void __CrossPlatformChatState__ChangeNotifyFunc();
void __bCrossPlatformEnabled__ChangeNotifyFunc();
bool IsCrossPlatformEnabled();
bool VerifyCrossPlatformChat();
bool VerifyCrossPlatformEnabled();
void ReadPlatformSettings();
void RecheckAllSettings();
void SavedCrossPlatformEnabledChanged();
void HandlePlatformSettingsChanged(class UPlatformAccountSettings* _);
void HandleNetworkSaveSet(class UNetworkSave_TA* InSave);
void HandlePlatformSettingsCreated(class UPlatformAccountSettings* InPlatformSettings,
uint8_t InControllerID);
void eventConstruct();
void EventAccountSettingsChanged(class UAccountSettingsComponent_TA* AccountSettings);
};

// Class TAGame.__OnlineGameParty_TA__HandleProfileSet_0x2
// 0x0008 (0x0060 - 0x0068)
class U__OnlineGameParty_TA__HandleProfileSet_0x2 : public UObject
{
public:
class ULocalPlayer_TA* P; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameParty_TA__HandleProfileSet_0x2");
}

return uClassPointer;
};

void __OnlineGameParty_TA__HandleProfileSet_0x3(class USaveData_TA* _);
void __OnlineGameParty_TA__HandleProfileSet_0x2();
};

// Class TAGame.__OnlineGameParty_TA__HandleTradingEnabledChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OnlineGameParty_TA__HandleTradingEnabledChanged_0x1 : public UObject
{
public:

```

```
class UGameplaySettingsSave_TA*          GameplaySettings;           // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._OnlineGameParty_TA__HandleTradingEnabledChanged_0x1");

}

return uClassPointer;
};

};

// Class TAGame._OnlineGameParty_TA__IsPrimaryMemberForPlayer_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__OnlineGameParty_TA__IsPrimaryMemberForPlayer_0x1 : public UObject
{
public:
struct FUniqueNetId           UserId;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._OnlineGameParty_TA__IsPrimaryMemberForPlayer_0x1");

}

return uClassPointer;
};

bool __OnlineGameParty_TA__IsPrimaryMemberForPlayer_0x1(struct FPartyMember P);

};

// Class TAGame._OnlineGameParty_TA__IsProductValidAfterVerify_0x1
// 0x0040 (0x0060 - 0x00A0)
class U__OnlineGameParty_TA__IsProductValidAfterVerify_0x1 : public UObject
{
public:
struct FOnlineProductData        OPD;           // 0x0060 (0x0040)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._OnlineGameParty_TA_IsProductValidAfterVerify_0x1");
}

return uClassPointer;
};

};

// Class
TAGame._OnlineGameParty_TA_RestoreArchivedAndFavoritedPendingProductOffers_0x1
// 0x0018 (0x0060 - 0x0078)
class U_OnlineGameParty_TA_RestoreArchivedAndFavoritedPendingProductOffers_0x1 :
public UObject
{
public:
class UProductsFavoriteSave_TA*           ProductsFavoriteSave;          // 0x0060
(0x0008) [0x0000000000000000]
class USaveData_TA*                      SaveData;                     // 0x0068 (0x0008)
[0x0000000000000000]
class UProductsArchiveSave_TA*           ArchiveSave;                 // 0x0070 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._OnlineGameParty_TA_RestoreArchivedAndFavoritedPendingProductOffers_0x1");
}

return uClassPointer;
};

};

// Class TAGame.ProductsHashSave_TA
// 0x0048 (0x00C8 - 0x0110)
class UProductsHashSave_TA : public USaveObject_TA
{
public:
TArray<struct FProductHashID>      HashIDs;                  // 0x00C8 (0x0010)
[0x0008004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<uint64_t>                   InstanceIDs;              // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceID>   InstanceIDs128;         // 0x00E8

```

```

(0x0010) [0x0000008000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __HashIDs__ChangeNotify;           // 0x00F8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductsHashSave_TA");
}

return uClassPointer;
};

struct FProductHashID __ProductsHashSave_TA__OnLoad_0x1(struct FProductInstanceID
InstanceID);
struct FProductInstanceID __ProductsHashSave_TA__UpdateInstanceIDs_0x1(struct
FProductHashSource P);
void __HashIDs__ChangeNotifyFunc();
void UpdateInstanceIDs(struct FProductHashID HashID, class USaveData_TA* SaveData,
unsigned long bSaved);
void OnOnlineProductRemoved(class UOnlineProduct_TA* OnlineProduct, class USaveData_TA*
SaveData);
void OnNewOnlineProduct(class UOnlineProduct_TA* OnlineProduct);
int32_t NumProducts();
void SetIsSaved(struct FProductHashID HashID, class USaveData_TA* SaveData, unsigned long
bSaved);
bool GetIsSaved(struct FProductHashID HashID);
void OnLoad();
};

// Class TAGame.ProductsArchiveSave_TA
// 0x0028 (0x0110 - 0x0138)
class UProductsArchiveSave_TA : public UProductsHashSave_TA
{
public:
TArray<struct FProductHashID>           ArchivedHashIDs;           // 0x0110
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventProductArchived__Delegate;           // 0x0120
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductsArchiveSave_TA");
}
}

```

```

return uClassPointer;
};

void __ProductsArchiveSave_TA__GetVersionDelegates_0x2(class UObject* SaveObj);
void __ProductsArchiveSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void SetIsArchived(struct FProductHashID HashID, class USaveData_TA* SaveData, unsigned long bArchived);
bool GetIsArchived(struct FProductHashID HashID);
void EventProductArchived(struct FProductHashID HashID);
};

// Class TAGame.ProductsFavoriteSave_TA
// 0x0010 (0x0110 - 0x0120)
class UProductsFavoriteSave_TA : public UProductsHashSave_TA
{
public:
TArray<int32_t> FavortitedHashes; // 0x0110 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductsFavoriteSave_TA");
}

return uClassPointer;
};

void __ProductsFavoriteSave_TA__GetVersionDelegates_0x2(class UObject* SaveObj);
void __ProductsFavoriteSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
bool IsProductFavorited(struct FProductHashID HashID);
void SetIsFavorited(struct FProductHashID HashID, class USaveData_TA* SaveData, unsigned long bValue);
};

// Class TAGame.__OnlineGameParty_TA__SendTradeToBackEnd_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameParty_TA__SendTradeToBackEnd_0x1 : public UObject
{
public:
TArray<struct FProductInstanceId> LocalProductOffers; // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameParty_TA__SendTradeToBackEnd_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__OnlineGameTourServer_TA__AllPlayersInGame_0x2
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameTourServer_TA__AllPlayersInGame_0x2 : public UObject
{
public:
TArray<struct FUniqueNetId>           AllPlayers;           // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameTourServer_TA__AllPlayersInGame_0x2");
}

return uClassPointer;
};

struct FUniqueNetId
__OnlineGameTourServer_TA__AllPlayersInGame_0x2__OnlineGameTourServer_TA__AllPlay
rsInGame_0x2_0x1(struct FTourPlayer P);
void __OnlineGameTourServer_TA__AllPlayersInGame_0x2(struct FTourTeam T);
};

// Class TAGame.OnlineGameTourServer_TA
// 0x0200 (0x00B0 - 0x02B0)
class UOnlineGameTourServer_TA : public UOnline_X
{
public:
struct FTourServerSettings           TourSettings;           // 0x00B0 (0x00E8)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FTourServerSettings           ReservationsTourSettings; // 0x0198
(0x00E8) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UMapSet_TA*                   MapSet;               // 0x0280 (0x0008)
[0x0001004000002000] (CPF_Transient)
class UTourGameUpdateDispatcher_TA*   GameUpdateDispatcher; // 0x0288 (0x0008) [0x0001004000002000] (CPF_Transient)
class UTourServerInfo_TA*           TourServerInfo;        // 0x0290 (0x0008)
[0x0001004000000000]

```

```

TArray<uint64_t> TeamIDs; // 0x0298 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTourServerConfig_TA* TournamentServerConfig; // 0x02A8
(0x0008) [0x0001800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameTourServer_TA");
}

return uClassPointer;
};

void __OnlineGameTourServer_TA__OnInit_0x1(class UIReservationConnection_X* Connection,
class UObject* Message);
uint64_t __OnlineGameTourServer_TA__InitTourSettings_0x1(struct FTourTeam T);
bool __OnlineGameTourServer_TA__AllPlayersInGame_0x4(struct FUniqueNetId P);
struct FUniqueNetId __OnlineGameTourServer_TA__AllPlayersInGame_0x3(struct FTourPlayer P);
bool __OnlineGameTourServer_TA__AllPlayersInGame_0x1(struct FTourTeam T);
bool __OnlineGameTourServer_TA__HasMinPlayersPerTeam_0x1(struct FTourTeam T);
bool __OnlineGameTourServer_TA__HasMinPlayersPerTeam_0x2(struct FTourPlayer P);
void __OnlineGameTourServer_TA__InitGameEventTeamScoresWhenInitialized_0x2(class ATeam_Soccar_TA* T);
void __OnlineGameTourServer_TA__InitGameEventTeamScoresWhenInitialized_0x1(struct FTourTeam T);
class UObject* CreateClientReservationMessage(class UReservationBeacon_X* Beacon, struct FServerReservationData& Reservation, struct FUniqueNetId& PlayerID);
void HandleJoinFacelitReservation(class UPsyNetService_JoinExternalMatch_X* Notification);
void HandleCreateFacelitReservation(class UPsyNetService_CreateExternalMatch_X*
Notification);
void SetTeamDifficultiesByTeamID(uint64_t TeamID, int32_t DifficultyValue);
int32_t GetTeamScore(int32_t TeamIndex);
void InitGameEventTeamScoresWhenInitialized(class AGameEvent_Soccar_TA* InGameEvent);
void InitGameEventTeamScores(class AGameEvent_Soccar_TA* InGameEvent);
bool IsBotOnlyTeam(class ATeam_TA* Team);
int32_t GetTeamIndexFromTeamID(uint64_t TeamID);
void InitMatchSeriesScores(class UMatchSeries_TA* MatchSeries);
void HandleActiveGameUpdated();
void HandleError(class UError* Error);
void HandleGameEventInitialized(class AGameEvent_Soccar_TA* InGameEvent);
bool IsMatchCancelled();
void GoToNextTourMap();
void AllowPlayerLogin(class FString Options, struct FUniqueNetId PlayerID, class FString& ErrorMessage);
bool AllowSplitscreenJoin(struct FUniqueNetId PrimaryPlayerId, struct FUniqueNetId PlayerID,
class FString PlayerName, class FString& Error);
int32_t GetNumAllowedSpectators();
int32_t GetTeamNum(struct FUniqueNetId& PlayerID);
bool IsPlayerRegistered(struct FUniqueNetId PlayerID);

```

```

bool AllPlayersRegistered(TArray<struct FReservationPlayerData>& Players);
bool CanAcceptReservations(class UAddReservationMessagePublic_X* Message);
static TArray<class UMapData_TA*> FilterMapData(int32_t GameMode, TArray<struct FName>& FilterMaps);
void UpdatePlayerTeam(class APlayerReplicationInfo* PRI, class AGameEvent_Team_TA* GameEventTeam);
bool HasMinPlayersPerTeam();
bool AllPlayersInGame(TArray<struct FUniqueNetId>& OutPlayersInGame);
bool IsBotTeam(uint64_t TeamID);
void UpdatePlayerReservations();
static struct FCustomMatchSettings CreateMatchSettings(struct FTourServerSettings InSettings,
class UTourServerInfo_TA* InTourServerInfo, class UTourServerConfig_TA*
InTournamentServerConfig);
static void EnsureDifferentTeamColors(struct FTourServerSettings& Settings);
void InitTourSettings(unsigned long bUpdateReservations, struct FTourServerSettings& NewSettings);
void HandleTourMigrationMessage(class UTourMatchInfoMessage_TA* Message);
void HandleJoinTournamentReservation(class UPsyNetService_JoinTournament_X*
Notification);
void HandleCreateTournamentReservation(class UPsyNetService_CreateTournament_X*
Notification);
bool IsPlaylistName(struct FName PlaylistName);
bool IsTournamentPendingOrSet();
bool IsTournamentSet();
bool IsTournamentPlaylist(int32_t PlaylistId);
bool HasTournamentPlaylist();
void ClearSettings();
void OnInit();
};

// Class TAGame.__OnlineGameTourServer_TA__FilterMapData_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlineGameTourServer_TA__FilterMapData_0x1 : public UObject
{
public:
TArray<struct FName> FilterMaps; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameTourServer_TA__FilterMapData_0x1");
}

return uClassPointer;
};

bool __OnlineGameTourServer_TA__FilterMapData_0x1(class UMapData_TA* M);
};

```

```
// Class TAGame.__OnlineGameTourServer_TA__GetTeamScore_0x1
// 0x0004 (0x0060 - 0x0064)
class U__OnlineGameTourServer_TA__GetTeamScore_0x1 : public UObject
{
public:
int32_t TeamIndex; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameTourServer_TA__GetTeamScore_0x1");
}

return uClassPointer;
};

bool __OnlineGameTourServer_TA__GetTeamScore_0x1(struct FTourMatchGame G);
};

// Class TAGame.__OnlineGameTourServer_TA__IsBotOnlyTeam_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OnlineGameTourServer_TA__IsBotOnlyTeam_0x1 : public UObject
{
public:
class ATeam_TA* Team; // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineGameTourServer_TA__IsBotOnlyTeam_0x1");
}

return uClassPointer;
};

bool __OnlineGameTourServer_TA__IsBotOnlyTeam_0x1(uint64_t T);
};

// Class TAGame.OnlinePlayerFriends_TA
// 0x0020 (0x0380 - 0x03A0)
class UOnlinePlayerFriends_TA : public UOnlinePlayerFriends_X
```

```
{  
public:  
class UPersonas_TA* PersonasData; // 0x0380 (0x0008)  
[0x0000804000002000] (CPF_Transient)  
struct FScriptDelegate __EventPsyNetRecentPlayers__Delegate; // 0x0388  
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.OnlinePlayerFriends_TA");  
}  
  
return uClassPointer;  
};  
  
void TryCombiningFriendsList(struct FOnlineFriend& PlatformPresence, struct FOnlineFriend& PreviousPlatformPresence, struct FOnlineFriend& EpicPresence, struct FOnlineFriend& PreviousEpicPresence);  
void HandleLinkedAccountsToBlockReceived(unsigned long bSuccess, TArray<struct FLinkedAccountData> RequestedAccounts);  
void HandleEpicPresenceChanged(struct FUniqueNetId EpicAccountId);  
void HandlePlatformPresenceChange(struct FUniqueNetId PlatformAccountId);  
void DownloadRecentPlayerDetails(struct FScriptDelegate Callback, TArray<struct FUniqueNetId>& RecentPlayerIds);  
void EventPsyNetRecentPlayers(TArray<struct FOnlineFriend>& RecentPlayerData);  
};  
  
// Class TAGame.__OnlinePlayerFriends_TA__DownloadRecentPlayerDetails_0x1  
// 0x0018 (0x0060 - 0x0078)  
class U__OnlinePlayerFriends_TA__DownloadRecentPlayerDetails_0x1 : public UObject  
{  
public:  
struct FScriptDelegate Callback; // 0x0060 (0x0018)  
[0x000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__OnlinePlayerFriends_TA__DownloadRecentPlayerDetails_0x1");  
}  
  
return uClassPointer;  
};
```

```

void __OnlinePlayerFriends_TA__DownloadRecentPlayerDetails_0x1(class
URPC_GetPersonalInfo_TA* R);
};

// Class TAGame.RPC_GetPersonalInfo_TA
// 0x0030 (0x00E8 - 0x0118)
class URPC_GetPersonalInfo_TA : public URPC_X
{
public:
TArray<struct FUniqueNetId> PlayerIds; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPsyNetPersonaData> PlayerData; // 0x00F8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FOnlineFriend> FriendsData; // 0x0108 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetPersonalInfo_TA");
}

return uClassPointer;
};

void eventOnSuccess();
class URPC_GetPersonalInfo_TA* SetPlayerIDs(TArray<struct FUniqueNetId> InPlayerIDs);
};

// Class TAGame.__OnlinePlayerMTX_TA__HandleClaimSuccess_0x1
// 0x0010 (0x0060 - 0x0070)
class U__OnlinePlayerMTX_TA__HandleClaimSuccess_0x1 : public UObject
{
public:
class USaveData_TA* SaveData; // 0x0060 (0x0008)
[0x0000000000000000]
class URPC_MicroTransactions_ClaimEntitlements_TA* RPC; // 0x0068
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlinePlayerMTX_TA__HandleClaimSuccess_0x1");
}
}

```

```

return uClassPointer;
};

void __OnlinePlayerMTX_TA__HandleClaimSuccess_0x1();
};

// Class TAGame.RPC_MicroTransactions_ClaimEntitlements_TA
// 0x0088 (0x00E8 - 0x0170)
class URPC_MicroTransactions_ClaimEntitlements_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class FString                 AuthCode;          // 0x0130 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FOnlineProductData> Products;      // 0x0140 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FCurrency>     Currencies;        // 0x0150 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FCurrency>     WalletCurrencies; // 0x0160 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.RPC_MicroTransactions_ClaimEntitlements_TA");
        }
    }

    return uClassPointer;
};

class URPC_MicroTransactions_ClaimEntitlements_TA* SetAuthCode(class FString InAuthCode);
class URPC_MicroTransactions_ClaimEntitlements_TA* SetPlayerID(struct FUniqueNetId
InPlayerId);
};

// Class TAGame.OnlinePlayerMTX_TA
// 0x0038 (0x0060 - 0x0098)
class UOnlinePlayerMTX_TA : public UObject
{
public:
    unsigned long           bEntitlementsDirty : 1;      // 0x0060 (0x0004)
    [0x0000000000000000] [0x00000001]
    unsigned long           bClaimingEntitlements : 1;    // 0x0060 (0x0004)
    [0x0000000000000000] [0x00000002]
    unsigned long           bMtxCodeExpired : 1;         // 0x0060 (0x0004)
    [0x0000000000000000] [0x00000004]
    unsigned long           bAllowEntitlements : 1;       // 0x0060 (0x0004)
    [0x00000000000000004002] [0x00000008] (CPF_Const | CPF_Config)
};

```

```

struct FScriptDelegate           __EventClaimedEntitlements__Delegate;      // 0x0068
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventEntitlementsError__Delegate;        // 0x0080
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlinePlayerMTX_TA");
}

return uClassPointer;
};

void HandleClaimFail(class URPC_MicroTransactions_ClaimEntitlements_TA* RPC);
void HandleEntitlements(class USaveData_TA* SaveData, class
URPC_MicroTransactions_ClaimEntitlements_TA* RPC);
void HandleClaimSuccess(class URPC_MicroTransactions_ClaimEntitlements_TA* RPC);
void HandleMtxCode(unsigned long bSuccess, class FString Code);
void ConditionalClaimEntitlements();
void ClaimEntitlements();
void HandlePsyNetDisconnected(class UPsyNetConnection_X* C);
void HandlePsyNetConnected(class UPsyNetConnection_X* C);
void eventConstruct();
void EventEntitlementsError(class UOnlinePlayerMTX_TA* MTX, class UError* Error);
void EventClaimedEntitlements(class UOnlinePlayerMTX_TA* MTX, TArray<class
UOnlineProduct_TA*> Products);
};

// Class TAGame.__OnlineProduct_TA__RemoveAttributes_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OnlineProduct_TA__RemoveAttributes_0x1 : public UObject
{
public:
class UClass*                  AttributeClass;                      // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineProduct_TA__RemoveAttributes_0x1");
}

return uClassPointer;
};

```

```
bool __OnlineProduct_TA__RemoveAttributes_0x1(class UProductAttribute_TA* A);
};

// Class TAGame.__OnlineProduct_TA__SortOnlineProductsByQuality_0x1
// 0x0004 (0x0060 - 0x0064)
class U__OnlineProduct_TA__SortOnlineProductsByQuality_0x1 : public UObject
{
public:
int32_t                                I;                                // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineProduct_TA__SortOnlineProductsByQuality_0x1");
}

return uClassPointer;
};

bool __OnlineProduct_TA__SortOnlineProductsByQuality_0x1(class UOnlineProduct_TA* P);

// Class TAGame.__OnlineProductExpirationHelper_TA__HandleNotificationSaveAdded_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OnlineProductExpirationHelper_TA__HandleNotificationSaveAdded_0x1 : public
UObject
{
public:
class UOnlineProduct_TA*                OnlineProduct;                // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineProductExpirationHelper_TA__HandleNotificationSaveAdded_0x1");
}

return uClassPointer;
};

bool __OnlineProductExpirationHelper_TA__HandleNotificationSaveAdded_0x1(class
UNotification_TA* N);
```

```

};

// Class TAGame.ProductExpirationNotification_TA
// 0x0030 (0x0170 - 0x01A0)
class UProductExpirationNotification_TA : public UNotification_TA
{
public:
    struct FProductHashID           HashID;                      // 0x0170 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    class FString                  ExpiredTitle;                // 0x0178 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                  ExpiredBody;                 // 0x0188 (0x0010)
    [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class UOnlineProduct_TA*      OnlineProduct;               // 0x0198 (0x0008)
    [0x0000004000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductExpirationNotification_TA");
        }

        return uClassPointer;
    }

    class UProductExpirationNotification_TA* SetOnlineProduct(class UOnlineProduct_TA*
InProduct);
    };

// Class TAGame.OnlineProductExpirationHelper_TA
// 0x0038 (0x0060 - 0x0098)
class UOnlineProductExpirationHelper_TA : public UObject
{
public:
    TArray<class UOnlineProduct_TA*>      ExpirationProducts;   // 0x0060
    (0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<class UOnlineProduct_TA*>      ExpiredProducts;     // 0x0070
    (0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<class UOnlineProduct_TA*>      ExpiredProductNotificationQueue; // 0x0080
    (0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
    class UOnlineProductStoreSet_TA*      ExpiredProductsSet;    // 0x0090
    (0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component |
CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.OnlineProductExpirationHelper_TA");
}

return uClassPointer;
};

void HandleAcceptNotification(class UProductExpirationNotification_TA* Notification);
void ExpireProduct(class UOnlineProduct_TA* InProduct);
void HandleNotificationSaveAdded(class UNotificationSave_TA* NotificationSave);
void CheckForExpiredProducts();
void FindNextExpirationTime();
void RemoveProduct(class UOnlineProduct_TA* InProduct);
void AddExpirationProduct(class UOnlineProduct_TA* InProduct);
void CheckProductForExpiration(class UOnlineProduct_TA* InProduct);
void ClearExpirationProducts();
void eventConstruct();
};

// Class TAGame.__OnlineProductStore_TA__FindProductsFromHash_0x1
// 0x0004 (0x0060 - 0x0064)
class U__OnlineProductStore_TA__FindProductsFromHash_0x1 : public UObject
{
public:
struct FProductHashID           HashID;                      // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineProductStore_TA__FindProductsFromHash_0x1");
}

return uClassPointer;
};

TArray<class UOnlineProduct_TA*>
__OnlineProductStore_TA__FindProductsFromHash_0x1(class UOnlineProductStoreSet_TA* Set);
};

// Class TAGame.OnlineProductStoreSet_TA
// 0x0050 (0x00C0 - 0x0110)
class UOnlineProductStoreSet_TA : public UIdenticalProductCache_TA
{
public:
class ULocalPlayer*             Player;                      // 0x00C0 (0x0008)
[0x0000004000000000]
struct FUniqueNetId             PlayerID;                   // 0x00C8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
};

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineProductStoreSet_TA");
}

return uClassPointer;
};

class UOnlineProduct_TA* AddProductData(struct FOnlineProductData& Data);
struct FUniqueNetId GetPlayerID();
void InitRemote(struct FUniqueNetId InPlayerId);
void InitLocal(class ULocalPlayer* InPlayer);
};

// Class TAGame.__OnlineStorageSyncManager_TA__SyncCloudData_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OnlineStorageSyncManager_TA__SyncCloudData_0x1 : public UObject
{
public:
class URPC_X* ProductsPlayerGet; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__OnlineStorageSyncManager_TA__SyncCloudData_0x1");
}

return uClassPointer;
};

void __OnlineStorageSyncManager_TA__SyncCloudData_0x1();
};

// Class TAGame.__OutOfWorldVolume_TA__Touch_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OutOfWorldVolume_TA__Touch_0x1 : public UObject
{
public:
class AActor* Other; // 0x0060 (0x0008)
[0x0001000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__OutOfWorldVolume_TA__Touch_0x1");
}

return uClassPointer;
};

bool __OutOfWorldVolume_TA__Touch_0x1(class UClass* TouchType);
};

// Class TAGame.OutOfWorldVolume_TA
// 0x0034 (0x02A4 - 0x02D8)
class AOutOfWorldVolume_TA : public AVolume
{
public:
TArray<class UClass*> TouchTypes; // 0x02A8 (0x0010)
[0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
uint8_t OutOfWorldType; // 0x02B8 (0x0001)
[0x0001000000000001] (CPF_Edit)
struct FScriptDelegate _EventActorOutOfWorld__Delegate; // 0x02C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OutOfWorldVolume_TA");
}

return uClassPointer;
};

bool GameEventsIsActive();
void eventUnTouch(class AActor* Other);
void eventTouch(class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector HitLocation, struct FVector HitNormal);
void EventActorOutOfWorld(class AOutOfWorldVolume_TA* OutOfWorldVolume, class AActor* Actor);
};

// Class TAGame.__OutOfWorldVolume_TA__UnTouch_0x1
// 0x0008 (0x0060 - 0x0068)
class U__OutOfWorldVolume_TA__UnTouch_0x1 : public UObject
{
public:
class AActor* Other; // 0x0060 (0x0008)
};

```

```
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__OutOfWorldVolume_TA__UnTouch_0x1");
}

return uClassPointer;
};

bool __OutOfWorldVolume_TA__UnTouch_0x1(class UClass* TouchType);
};

// Class TAGame.__PerfStatGraph_TA__UpdateGraphRanges_0x1
// 0x0004 (0x0060 - 0x0064)
class U_PerfStatGraph_TA_UpdateGraphRanges_0x1 : public UObject
{
public:
float TargetMS; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PerfStatGraph_TA__UpdateGraphRanges_0x1");
}

return uClassPointer;
};

void __PerfStatGraph_TA__UpdateGraphRanges_0x1(class USampleHistory_TA* History);
};

// Class TAGame.__PlatformMetrics_TA__DispatchAndRemove_0x1
// 0x0008 (0x0060 - 0x0068)
class U_PlatformMetrics_TA_DispatchAndRemove_0x1 : public UObject
{
public:
class APlayerController_X* InPC; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PlatformMetrics_TA__DispatchAndRemove_0x1");
}

return uClassPointer;
};

bool __PlatformMetrics_TA__DispatchAndRemove_0x1(struct FPlatformMetric M);
};

// Class TAGame.PlatformMetrics_TA
// 0x0018 (0x0080 - 0x0098)
class UPlatformMetrics_TA : public UMetricsGroup_X
{
public:
class AGameEvent_Soccar_TA*           SoccarGame;          // 0x0080
(0x0008) [0x0000800000000000]
TArray<struct FPlatformMetric>         Metrics;            // 0x0088 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlatformMetrics_TA");
}

return uClassPointer;
};

void RecordTypes(struct FUniqueNetId InID, TArray<struct FPlatformMetricData> InMetricData);
void DispatchAndRemove(class APlayerController_X* InPC);
void HandleSoccarGameEnded(class AGameEvent_TA* _);
void HandleSoccarGameInjected();
void SetType(class APlayerController_TA* InPC, int32_t InType);
};

// Class TAGame.__PlatformMetrics_TA__SetType_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlatformMetrics_TA_SetType_0x1 : public UObject
{
public:
class APlayerController_TA*           InPC;               // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__PlatformMetrics_TA__SetType_0x1");

}

return uClassPointer;
};

bool __PlatformMetrics_TA__SetType_0x1(struct FPlatformMetric M);

// Class TAGame.__PlayerController_TA__ClientNotifyChallengedToHonorDuel_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__PlayerController_TA__ClientNotifyChallengedToHonorDuel_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;                      // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PlayerController_TA__ClientNotifyChallengedToHonorDuel_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__PlayerController_TA__ClientNotifyHonorDuelWithPlayers_0x1
// 0x0090 (0x0060 - 0x00F0)
class U__PlayerController_TA__ClientNotifyHonorDuelWithPlayers_0x1 : public UObject
{
public:
struct FUniqueNetId           Challenger;                  // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FUniqueNetId           Defender;                    // 0x00A8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class
TAGame.__PlayerController_TA__ClientNotifyHonorDuelWithPlayers_0x1");
}

return uClassPointer;
};

};

// Class TAGame.__PlayerController_TA__HandleProfileGamepadSave_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlayerController_TA__HandleProfileGamepadSave_0x1 : public UObject
{
public:
class UProfileGamepadSave_TA*           GamepadSave;          // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PlayerController_TA__HandleProfileGamepadSave_0x1");
}

return uClassPointer;
};

void __PlayerController_TA__HandleProfileGamepadSave_0x1(class APRI_TA* InPRI);

};

// Class TAGame.ProfileGamepadSave_TA
// 0x0020 (0x00C8 - 0x00E8)
class UProfileGamepadSave_TA : public USaveObject_TA
{
public:
TArray<struct FPlayerBinding>           GamepadBindings;      // 0x00C8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
float                                     ControllerDeadzone;   // 0x00D8 (0x0004)
[0x0000000000000000]
float                                     DodgeInputThreshold; // 0x00DC (0x0004)
[0x0000000000000000]
float                                     SteeringSensitivity; // 0x00E0 (0x0004)
[0x0000000000000000]
float                                     AirControlSensitivity; // 0x00E4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfileGamepadSave_TA");
}

return uClassPointer;
};

void __ProfileGamepadSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void OnCreate();
void SetAirControlSensitivity(float Value);
void SetSteeringSensitivity(float Value);
void SetDodgeInputThreshold(float Value);
};

// Class TAGame.__PlayerController_TA__InitNetworkSave_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlayerController_TA__InitNetworkSave_0x1 : public UObject
{
public:
class UNetworkSave_TA*           NetworkSave;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PlayerController_TA__InitNetworkSave_0x1");
}

return uClassPointer;
};

void __PlayerController_TA__InitNetworkSave_0x1();
};

// Class TAGame.NetworkSave_TA
// 0x0038 (0x00C8 - 0x0100)
class UNetworkSave_TA : public USaveObject_TA
{
public:
unsigned long           bEnableCrossPlatform : 1;           // 0x00C8 (0x0004)
[0x0008000000000000] [0x00000001]
unsigned long           bDisableCrossPlay : 1;           // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000002]
uint8_t                IncomingPacketFrequency;           // 0x00CC (0x0001)
[0x0000000000000000]

```

```
uint8_t OutgoingPacketFrequency; // 0x00CD (0x0001)
[0x0000000000000000]
uint8_t IncomingBandwidth; // 0x00CE (0x0001)
[0x0000000000000000]
uint8_t NetworkInputBuffer; // 0x00CF (0x0001)
[0x0008000000000000]
struct FScriptDelegate __bEnableCrossPlatform__ChangeNotify; // 0x00D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __NetworkInputBuffer__ChangeNotify; // 0x00E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NetworkSave_TA");
}

return uClassPointer;
};

void __NetworkInputBuffer__ChangeNotifyFunc();
void __bEnableCrossPlatform__ChangeNotifyFunc();
void Deprecated_SetCrossplayDisabled(unsigned long bnewValue);
bool Deprecated_GetCrossplayDisabled();
int32_t GetOutgoingPacketFrequencyValue();
int32_t GetIncomingBandwidthValue();
int32_t GetIncomingPacketFrequencyValue();
void ApplySettings();
};

// Class TAGame.__PlayerControllerBase_TA__InitFromProfile_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlayerControllerBase_TA__InitFromProfile_0x1 : public UObject
{
public:
class UProfile_TA* Profile; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PlayerControllerBase_TA__InitFromProfile_0x1");
}

return uClassPointer;
}
```

```
};

void __PlayerControllerBase_TA__InitFromProfile_0x1(class UGFxShell_X* Shell);
};

// Class TAGame.__PlayerControllerBase_TA__Say_TA_0x3
// 0x0028 (0x0060 - 0x0088)
class U__PlayerControllerBase_TA__Say_TA_0x3 : public UObject
{
public:
    class UOnlinePlayer_TA*           OnlinePlayer;          // 0x0060 (0x0008)
    [0x0000000000000000]
    class UOnlineGame_TA*            OnlineGame;           // 0x0068 (0x0008)
    [0x0000000000000000]
    class UOnlineGameParty_TA*       OnlineGameParty;      // 0x0070
    (0x0008) [0x0000000000000000]
    TArray<struct FUniqueNetId>     AllowedPlayers;      // 0x0078 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__PlayerControllerBase_TA__Say_TA_0x3");
        }

        return uClassPointer;
    }

void __PlayerControllerBase_TA__Say_TA_0x8(struct FUniqueNetId P);
void __PlayerControllerBase_TA__Say_TA_0x7(struct FUniqueNetId P);
bool __PlayerControllerBase_TA__Say_TA_0x4(struct FUniqueNetId P);
bool __PlayerControllerBase_TA__Say_TA_0x3(struct FUniqueNetId P);
};

// Class TAGame.__PlayerInput_TA__RemoveBindingsForKey_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlayerInput_TA__RemoveBindingsForKey_0x1 : public UObject
{
public:
    struct FName                   KeyName;               // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
```

```

TAGame.__PlayerInput_TA_RemoveBindingsForKey_0x1");
}

return uClassPointer;
};

};

// Class TAGame.PlayerInput_TA
// 0x0050 (0x0478 - 0x04C8)
class UPlayerInput_TA : public UPlayerInput_X
{
public:
float GamepadLookScale; // 0x0478 (0x0004)
[0x0000000000004000] (CPF_Config)
float aTargetSelect; // 0x047C (0x0004)
[0x0000000000000004] (CPF_Input)
float TargetSelectDeadZone; // 0x0480 (0x0004)
[0x0000000000000000]
unsigned long bTargetSelected : 1; // 0x0484 (0x0004)
[0x0000000000000001]
unsigned long bIsUsingMouseSteer : 1; // 0x0484 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long bIsUsingMouseThrottle : 1; // 0x0484 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
unsigned long bIsUsingPositiveMouseAirYaw : 1; // 0x0484
(0x0004) [0x0000000000002000] [0x00000008] (CPF_Transient)
unsigned long bIsUsingNegativeMouseAirYaw : 1; // 0x0484
(0x0004) [0x0000000000002000] [0x00000010] (CPF_Transient)
unsigned long bIsUsingPositiveMouseAirPitch : 1; // 0x0484 (0x0004)
[0x0000000000002000] [0x00000020] (CPF_Transient)
unsigned long bIsUsingNegativeMouseAirPitch : 1; // 0x0484
(0x0004) [0x0000000000002000] [0x00000040] (CPF_Transient)
unsigned long bIsUsingPositiveMouseAirRoll : 1; // 0x0484 (0x0004)
[0x0000000000002000] [0x00000080] (CPF_Transient)
unsigned long bIsUsingNegativeMouseAirRoll : 1; // 0x0484 (0x0004)
[0x0000000000002000] [0x00000100] (CPF_Transient)
float aPositiveYaw; // 0x0488 (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
float aNegativeYaw; // 0x048C (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
float aPositivePitch; // 0x0490 (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
float aNegativePitch; // 0x0494 (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
float aPositiveRoll; // 0x0498 (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
float aNegativeRoll; // 0x049C (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
float aLookRoll; // 0x04A0 (0x0004)
[0x0000000000002004] (CPF_Input | CPF_Transient)
uint8_t bRoll; // 0x04A4 (0x0001)
[0x0000000000002004] (CPF_Input | CPF_Transient)
uint8_t bPitch; // 0x04A5 (0x0001)

```

```
[0x0000000000000004] (CPF_Input | CPF_Transient)
float aLookDown; // 0x04A8 (0x0004)
[0x0000000000000004] (CPF_Input)
float aTurnRight; // 0x04AC (0x0004)
[0x0000000000000004] (CPF_Input)
float aTurnLeft; // 0x04B0 (0x0004)
[0x0000000000000004] (CPF_Input)
float aMouseForward; // 0x04B4 (0x0004)
[0x0000000000000004] (CPF_Input)
float aMouseStrafe; // 0x04B8 (0x0004)
[0x0000000000000004] (CPF_Input)
float aDodgeForward; // 0x04BC (0x0004)
[0x0000000000000004] (CPF_Input)
class UProfile_TA* Profile; // 0x04C0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_TA");
}

return uClassPointer;
};

void __PlayerInput_TA__Construct_0x2(class UPlayerInput_X* _);
void __PlayerInput_TA__Construct_0x1(class UPlayerInput_X* _);
static TArray<struct FBindingAction> GetDefaultActions();
void UpdateTargetSelect(class APlayerController_TA* PC);
struct FName GetKeyForAction(struct FName Action, unsigned long bGamepad, unsigned long bIncludeDisabledActions);
bool IsActionBound(struct FName Action);
class FString GetUIKeyForAction(struct FName Action, unsigned long bGamepad, unsigned long bIncludeDisabledActions);
bool IsUsingMouseInput(struct FName Action, struct FName Key);
void RemoveBindingsFromArray(struct FName Category, TArray<struct FPlayerBinding>& OutBindings);
void RemoveBindings(struct FName Category);
bool IsKeyPressed(struct FName Key);
void eventPlayerInput(float DeltaTime);
void BuildBindings(TArray<struct FPlayerBinding>& PC, TArray<struct FPlayerBinding>& Gamepad);
void UpdateControlPreset();
void HandleProfileControlsSave(class UProfileControlsSave_TA* ControlsSave);
void OnActiveBindingsChanged();
void HandleSetBindingsToUserBindings(class UPlayerInput_X* PlayerInput);
void HandleGameplaySettingsLoaded(class UGameplaySettingsSave_TA*
GameplaySettingsSave);
void HandleProfileGamepadSave(class UProfileGamepadSave_TA* GamepadSave);
void HandleProfilePCSave(class UProfilePCSave_TA* ProfilePCSave);
```

```
void InitFromProfile(class UProfile_TA* InProfile);
void UpdateControlBindings();
void ShutdownInputSystem();
void eventConstruct();
};

// Class TAGame.__PlayerSpawnSystem_TA__CreateSpawnTicket_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlayerSpawnSystem_TA__CreateSpawnTicket_0x1 : public UObject
{
public:
class AGameEvent_TA* GameEvent; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PlayerSpawnSystem_TA__CreateSpawnTicket_0x1");
}

return uClassPointer;
};

void __PlayerSpawnSystem_TA__CreateSpawnTicket_0x1(class AController* InC);
};

// Class TAGame.PlayerSpawnSystem_TA
// 0x0000 (0x0060 - 0x0060)
class UPlayerSpawnSystem_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerSpawnSystem_TA");
}

return uClassPointer;
};

static void CreateReplicatedSpawnTicket(class AController* C, float RespawnDelaySeconds);
static void CreateSpawnTicket(class AController* C, class APRI_TA* PRI, class AGameEvent_TA*
GameEvent, float RespawnDelaySeconds);
static void SetupTicketHudCallbacks(class AReplicatedRespawnTicket_TA* Ticket, class
```

```

AGFxHUD_TA* HUD);
static void HandleReplicatedRespawnTicketRemoved(class AReplicatedRespawnTicket_TA*
Ticket, class AGFxHUD_TA* HUD);
static void TickRespawnDisplay(class UGameTick* Tick, class AReplicatedRespawnTicket_TA*
Ticket, class AGFxHUD_TA* HUD);
static void HandleCarDemolished(class UPlayerSpawnFeature_TA* Feature, class AController* C,
class APRI_TA* PRI, class UDemolishedCar_TA* Car);
static void HandlePlayerSpawned(class UPlayerSpawnFeature_TA* Feature, class AController* C,
class APawn* P);
static void TickSpawn(class UGameTick* Tick, class UPlayerSpawnFeature_TA* Feature, class
UPlayerSpawnTicket_TA* Ticket);
static void HandleBotReplacementCleared(class AReplicatedRespawnTicket_TA* Ticket, class
UBotReplacement_TA* Replacement);
static void HandleBotReplacementSet(class UPlayerSpawnFeature_TA* Feature, class
UPlayerSpawnTicket_TA* Ticket, class UBotReplacement_TA* Replacement);
static void HandlePlayerSpawnStop(class UPlayerSpawnFeature_TA* Feature, class AController* C);
static void HandlePlayerSpawnStart(class UPlayerSpawnFeature_TA* Feature, class AController* C,
class APRI_TA* PRI, class UPlayer* InPlayer, class AGameEvent_TA* GameEvent);
static void HandleSpawnStop(class UPlayerSpawnState_TA* State, class
UPlayerSpawnFeature_TA* Feature);
static void HandleSpawnStart(class UPlayerSpawnConfig_TA* Config, class
UPlayerSpawnState_TA* State, class AGameEvent_TA* GameEvent);
};

// Class TAGame.__PlayerVanity_TA__EquipProduct_0x1
// 0x0020 (0x0060 - 0x0080)
class U__PlayerVanity_TA__EquipProduct_0x1 : public UObject
{
public:
    struct FProductHashSource           ProductSource;          // 0x0060 (0x0020)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__PlayerVanity_TA__EquipProduct_0x1");
        }

        return uClassPointer;
    }

    void __PlayerVanity_TA__EquipProduct_0x1(struct FAssetLoadResult Result);
};

// Class TAGame.__PlaylistSkillDataSave_TA__Reconcile_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PlaylistSkillDataSave_TA__Reconcile_0x1 : public UObject
{
public:

```

```

class UPlaylistSkillDataSave_TA*           Other;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._PlaylistSkillDataSave_TA__Reconcile_0x1");
}

return uClassPointer;
};

bool __PlaylistSkillDataSave_TA__Reconcile_0x1(struct FPlaylistSkillData S);
};

// Class TAGame.PlaylistSkillDataSave_TA
// 0x0010 (0x00C8 - 0x00D8)
class UPlaylistSkillDataSave_TA : public USaveObject_TA
{
public:
TArray<struct FPlaylistSkillData>      SkillData;          // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlaylistSkillDataSave_TA");
}

return uClassPointer;
};

bool CompareMatchesPlayed(struct FPlaylistSkillData LocalSkill, TArray<struct
FPlaylistSkillData>& RemoteSkillData);
class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
};

// Class TAGame.__PrivacyPolicySave_TA__FindPolicyIndex_0x1
// 0x0008 (0x0060 - 0x0068)
class U__PrivacyPolicySave_TA__FindPolicyIndex_0x1 : public UObject
{
public:
int32_t                                InTextHash;          // 0x0060 (0x0004)
[0x0000000000000000]
int32_t                                InVersion;          // 0x0064 (0x0004)

```

```

[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__PrivacyPolicySave_TA__FindPolicyIndex_0x1");
}

return uClassPointer;
};

bool __PrivacyPolicySave_TA__FindPolicyIndex_0x1(struct FLegalTextVersion P);
};

// Class TAGame.PrivacyPolicySave_TA
// 0x0010 (0x00C8 - 0x00D8)
class UPrivacyPolicySave_TA : public USaveObject_TA
{
public:
TArray<struct FLegalTextVersion> AcceptedPrivatePolicies; // 0x00C8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PrivacyPolicySave_TA");
}

return uClassPointer;
};

int32_t FindPolicyIndex(int32_t InTextHash, int32_t InVersion);
void Accepted(int32_t InTextHash, int32_t InVersion);
bool RequiresAcceptance(int32_t InTextHash, int32_t InVersion);
};

// Class TAGame.__ProductAsset_Attachment_TA__ModifyThumbnailScene_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ProductAsset_Attachment_TA__ModifyThumbnailScene_0x1 : public UObject
{
public:
class USkeletalMeshComponent* AttachSKC; // 0x0060
(0x0008) [0x00000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._ProductAsset_Attachment_TA_ModifyThumbnailScene_0x1");
}

return uClassPointer;
};

void __ProductAsset_Attachment_TA_ModifyThumbnailScene_0x1(class
UAttachmentBehavior_TA* X);
};

// Class TAGame._ProductAsset_Skin_TA_AttemptApplyChassisOverride_0x1
// 0x0004 (0x0060 - 0x0064)
class U_ProductAsset_Skin_TA_AttemptApplyChassisOverride_0x1 : public UObject
{
public:
int32_t ForBodyID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._ProductAsset_Skin_TA_AttemptApplyChassisOverride_0x1");
}

return uClassPointer;
};

bool __ProductAsset_Skin_TA_AttemptApplyChassisOverride_0x1(struct FBodyChassisOverride
Override);
};

// Class TAGame.ProductOverride_SpecificMaterials_TA
// 0x0010 (0x0060 - 0x0070)
class UProductOverride_SpecificMaterials_TA : public UProductOverride_TA
{
public:
TArray<struct FMaterialAndMaterialOverride> MaterialOverrides; // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_SpecificMaterials_TA");
}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);
};

// Class TAGame.__ProductAsset_Skin_TA__GetSkinBodySettings_0x1
// 0x0004 (0x0060 - 0x0064)
class U__ProductAsset_Skin_TA__GetSkinBodySettings_0x1 : public UObject
{
public:
int32_t ForBodyID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductAsset_Skin_TA__GetSkinBodySettings_0x1");
}

return uClassPointer;
};

bool __ProductAsset_Skin_TA__GetSkinBodySettings_0x1(struct FSkinBodySettings Setting);

// Class TAGame.__ProductAsset_Skin_TA__GetSkinParameters_0x1
// 0x0004 (0x0060 - 0x0064)
class U__ProductAsset_Skin_TA__GetSkinParameters_0x1 : public UObject
{
public:
int32_t ForBodyID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class

```

```

TAGame.__ProductAsset_Skin_TA__GetSkinParameters_0x1");
}

return uClassPointer;
};

bool __ProductAsset_Skin_TA__GetSkinParameters_0x1(struct FSkinBodySettings Setting);
};

// Class TAGame.__ProductAttribute_ModifyTransformationPerBody_TA__ApplyToObject_0x1
// 0x0004 (0x0060 - 0x0064)
class U__ProductAttribute_ModifyTransformationPerBody_TA__ApplyToObject_0x1 : public
UObject
{
public:
int32_t BodyID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductAttribute_ModifyTransformationPerBody_TA__ApplyToObject_0x1");
}

return uClassPointer;
};

bool __ProductAttribute_ModifyTransformationPerBody_TA__ApplyToObject_0x1(class
UProductAssetReferenceBody_TA* X);
};

// Class TAGame.ProductAttribute_ModifyTransformationPerBody_TA
// 0x0038 (0x0080 - 0x00B8)
class UProductAttribute_ModifyTransformationPerBody_TA : public UProductAttribute_TA
{
public:
struct FVector LocationToAdd; // 0x0080 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator RotationToAdd; // 0x008C (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector ScaleToAdd; // 0x0098 (0x000C)
[0x0000000000000001] (CPF_Edit)
TArray<class UProductAssetReferenceBody_TA*> BodiesToApplyTo; // 
0x00A8 (0x0010) [0x000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_ModifyTransformationPerBody_TA");
}

return uClassPointer;
};

class UCarMeshComponent_TA* GetCarMesh(class UObject* Target);
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
};

// Class TAGame.__ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x1
// 0x0020 (0x0060 - 0x0080)
class U__ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x1 : public UObject
{
public:
class UProductPaint_TA*           Paint;           // 0x0060 (0x0008)
[0x0000000000000000]
class UObject*                     Target;         // 0x0068 (0x0008)
[0x0000000000000000]
class UCarMeshComponent_TA*       CarMesh;        // 0x0070 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UMaterialInstanceConstant*   SkinMaterial;    // 0x0078 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x1");
}

return uClassPointer;
};

void __ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x5(class
UProductOverride_TA* Override);
bool __ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x4(class
UProductOverride_TA* Override);
void __ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x3(class
UProductOverride_TA* PaintOverride);
bool __ProductAttribute_Painted_TA__ApplyPaintOverridesToObject_0x1(struct
FPaintWithOverride O);
};

// Class TAGame.__ProductAttribute_Painted_TA__ApplyToSetParameter_0x2
// 0x0020 (0x0060 - 0x0080)

```

```
class U__ProductAttribute_Painted_TA__ApplyToSetParameter_0x2 : public UObject
{
public:
    class UISetParameter*           Target_Object;          // 0x0060 (0x0008)
    [0x0000000000000000]
    class UISetParameter*           Target_Interface;      // 0x0068 (0x0008)
    [0x0000000000000000]
    class UProductAttribute_PaintSettings_TA*   PaintSettings;        // 0x0070
    (0x0008) [0x0000000000000000]
    class UProductPaint_TA*         Paint;                // 0x0078 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__ProductAttribute_Painted_TA__ApplyToSetParameter_0x2");
        }

        return uClassPointer;
    };

    void __ProductAttribute_Painted_TA__ApplyToSetParameter_0x2(struct
FPaintAttributeParameter PaintParam);
};

// Class TAGame.__ProductAttribute_UnlockForcedProducts_TA__ContainsSlotToUnlock_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ProductAttribute_UnlockForcedProducts_TA__ContainsSlotToUnlock_0x1 : public
UObject
{
public:
    class UProductEquipProfileSlot_ForceProduct_TA* SlotToFind;          // 0x0060
    (0x0008) [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__ProductAttribute_UnlockForcedProducts_TA__ContainsSlotToUnlock_0x1");
        }

        return uClassPointer;
    };

    bool __ProductAttribute_UnlockForcedProducts_TA__ContainsSlotToUnlock_0x1(class
```

```
UProductEquipProfileSlot_ForceProduct_TA* X);
};

// Class TAGame.__ProductLoader_TA__GetAssetByID_0x1
// 0x0004 (0x0060 - 0x0064)
class U__ProductLoader_TA__GetAssetByID_0x1 : public UObject
{
public:
int32_t ProductID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__ProductLoader_TA__GetAssetByID_0x1");
}

return uClassPointer;
};

bool __ProductLoader_TA__GetAssetByID_0x1(class UProductAsset_TA* A);
};

// Class TAGame.__ProductLoader_TA__GetAssetBySlot_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ProductLoader_TA__GetAssetBySlot_0x1 : public UObject
{
public:
class UProductSlot_TA* Slot; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductLoader_TA__GetAssetBySlot_0x1");
}

return uClassPointer;
};

bool __ProductLoader_TA__GetAssetBySlot_0x1(class UProductAsset_TA* A);
};

// Class TAGame.__ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x1
// 0x0008 (0x0060 - 0x0068)
```

```

class U__ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x1 : public
UObject
{
public:
class UMaterialInstanceConstant*           MIC;                      // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x1");
}

return uClassPointer;
};

void __ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x4(struct
FVectorParameterValue Param);
void __ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x3(struct
FTextureParameterValue Param);
void __ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x2(struct
FScalarParameterValue Param);
void __ProductOverride_MaterialParameter_TA__ForceApplyOverrideToMIC_0x1(struct
FFontParameterValue Param);
};

// Class TAGame.__ProductOverride_ParticleSystemColorParameter_TA__ApplyToObject_0x1
// 0x0018 (0x0060 - 0x0078)
class U__ProductOverride_ParticleSystemColorParameter_TA__ApplyToObject_0x1 : public
UObject
{
public:
class UISetParameter*           ParamTarget_Object;          // 0x0060 (0x0008)
[0x0000000000000000]
class UISetParameter*           ParamTarget_Interface;        // 0x0068 (0x0008)
[0x0000000000000000]
class UProductAttribute_PaintSettings_TA*   PaintSetting;        // 0x0070
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductOverride_ParticleSystemColorParameter_TA__ApplyToObject_0x1");
}
}

```

```

return uClassPointer;
};

void __ProductOverride_ParticleSystemColorParameter_TA__ApplyToObject_0x1(struct
FPaintColorParameterOverride O);
};

// Class TAGame.ProductOverride_ParticleSystemColorParameter_TA
// 0x0018 (0x0060 - 0x0078)
class UProductOverride_ParticleSystemColorParameter_TA : public UProductOverride_TA
{
public:
TArray<struct FPaintColorParameterOverride> ParameterOverrides; // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bUseParentEmissiveMultiplier : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
float PaintEmissiveMultiplier; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_ParticleSystemColorParameter_TA");
}

return uClassPointer;
};

struct FLinearColor GetPaintToApply(class UProductAttribute_PaintSettings_TA* PaintSetting,
struct FPaintColorParameterOverride& CurPaintParam);
class UProductAttribute_PaintSettings_TA* GetPaintSettings();
void ApplyToObject(class UObject* Target);
};

// Class TAGame.__ProductOverride_TA__ApplyOverrides_0x1
// 0x0008 (0x0060 - 0x0068)
class U__ProductOverride_TA__ApplyOverrides_0x1 : public UObject
{
public:
class UObject* Target; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class
TAGame._ProductOverride_TA__ApplyOverrides_0x1");
}

return uClassPointer;
};

void __ProductOverride_TA__ApplyOverrides_0x1(class UProductOverride_TA* O);
};

// Class TAGame._ProductsConfig_TA__IsTradeInAllowed_0x1
// 0x0004 (0x0060 - 0x0064)
class U__ProductsConfig_TA__IsTradeInAllowed_0x1 : public UObject
{
public:
int32_t SeriesID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._ProductsConfig_TA__IsTradeInAllowed_0x1");
}

return uClassPointer;
};

bool __ProductsConfig_TA__IsTradeInAllowed_0x1(struct FProductTradeInFilter P);
};

// Class TAGame.ProductsConfig_TA
// 0x0050 (0x0078 - 0x00C8)
class UProductsConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FReplacementProduct> ProhibitedProducts; // 0x0078
(0x0010) [0x0000008000400000] (CPF_NeedCtorLink)
TArray<struct FProductTradeInFilter> TradeInFilters; // 0x0088 (0x0010)
[0x001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FSeriesGroup> ItemSeries_AllowTradeIn; // 0x0098
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FSeriesGroup> ItemSeries_PreventP2PTrade; // 0x00A8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductTradeHoldOverride> ProductTradeHoldOverrides; // 0x00B8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductsConfig_TA");
}

return uClassPointer;
};

bool __ProductsConfig_TA__Apply_0x2(struct FProductTradeHoldOverride P);
bool __ProductsConfig_TA__Apply_0x1(struct FReplacementProduct P);
bool IsPlayerTradeProhibited(int32_t SeriesID, uint8_t ProductQuality);
struct FProductTradeInFilter GetTradeInFilter(int32_t FilterID);
bool IsTradeInAllowed(int32_t SeriesID, uint8_t ProductQuality);
class UOnlineProduct_TA* GetReplacementOnlineProduct(int32_t ProductID);
bool CanShowForOwningClient(int32_t ProductID);
bool IsProductIDProhibited(int32_t ProductID);
void Undo();
void Apply();
};

// Class TAGame.__ProductsHashSave_TA__UpdateInstanceIDs_0x2
// 0x0010 (0x0060 - 0x0070)
class U__ProductsHashSave_TA__UpdateInstanceIDs_0x2 : public UObject
{
public:
TArray<struct FProductInstanceId> Instances; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductsHashSave_TA__UpdateInstanceIDs_0x2");
}

return uClassPointer;
};

bool __ProductsHashSave_TA__UpdateInstanceIDs_0x2(struct FProductInstanceId Id);
};

// Class TAGame.__ProductTransactions_TA__GetEquippedProducts_0x2
// 0x0010 (0x0060 - 0x0070)
class U__ProductTransactions_TA__GetEquippedProducts_0x2 : public UObject
{
public:
TArray<struct FProductInstanceId> AllEquippedInstanceIDs; // 0x0060

```

(0x0010) [0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._ProductTransactions_TA__GetEquippedProducts_0x2");  
}  
  
return uClassPointer;  
};  
  
bool __ProductTransactions_TA__GetEquippedProducts_0x2(class UOnlineProduct_TA*  
OnlineProductEquipped);  
};  
  
// Class TAGame.ProductTransactions_TA  
// 0x0010 (0x0070 - 0x0080)  
class UProductTransactions_TA : public UComponent  
{  
public:  
class UOnlineProductStoreSet_TA*           InventoryProductSet;          // 0x0070  
(0x0008) [0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)  
class UOnlineProductStoreSet_TA*           TransactionProductSet;       // 0x0078  
(0x0008) [0x0000004004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductTransactions_TA");  
}  
  
return uClassPointer;  
};  
  
struct FProductInstanceID __ProductTransactions_TA__GetEquippedProducts_0x1(class  
UOnlineProduct_TA* EquippedProduct);  
struct FProductInstanceID __ProductTransactions_TA__GetUnequippedProducts_0x1(class  
UOnlineProduct_TA* EquippedProduct);  
TArray<class UOnlineProduct_TA*> GetTransactionProducts();  
TArray<struct FProductInstanceID> GetAllTransactionIDs();  
void ResetTransactionProducts();  
void Remove(class UOnlineProduct_TA* Product);  
void AddOnlineProduct(class UOnlineProduct_TA* OnlineProduct);  
void ClearOnlineProducts();  
void Unload();
```

```

TArray<class UOnlineProduct_TA*> GetUnequippedProducts(TArray<class UOnlineProduct_TA*>& OnlineProductSet, TArray<class UOnlineProduct_TA*>& AllEquippedProducts);
TArray<class UOnlineProduct_TA*> GetEquippedProducts(TArray<class UOnlineProduct_TA*>& OnlineProductSet, TArray<class UOnlineProduct_TA*>& AllEquippedProducts);
void RemoveQuantity(struct FProductHashID HashID, int32_t Quantity, TArray<class UOnlineProduct_TA*> AllEquippedProducts);
void AddQuantity(struct FProductHashID HashID, int32_t Quantity, TArray<class UOnlineProduct_TA*> AllEquippedProducts);
bool SetTransactionQuantity(struct FProductHashID HashID, int32_t Quantity, TArray<class UOnlineProduct_TA*> AllEquippedProducts);
void SetOnlineProducts(TArray<class UOnlineProduct_TA*>& InOnlineProducts);
void Init();
};

// Class TAGame.__ProductTransactions_TA__GetUnequippedProducts_0x2
// 0x0010 (0x0060 - 0x0070)
class U__ProductTransactions_TA__GetUnequippedProducts_0x2 : public UObject
{
public:
TArray<struct FProductInstanceId>           AllEquippedInstanceIDs;          // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ProductTransactions_TA__GetUnequippedProducts_0x2");
}

return uClassPointer;
};

bool __ProductTransactions_TA__GetUnequippedProducts_0x2(class UOnlineProduct_TA*
OnlineProduct);
};

// Class TAGame.__Profile_TA__ValidateLoadoutsWithProduct_0x1
// 0x0010 (0x0060 - 0x0070)
class U__Profile_TA__ValidateLoadoutsWithProduct_0x1 : public UObject
{
public:
struct FProductInstanceId           OnlinID;           // 0x0060 (0x0010)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Profile_TA__ValidateLoadoutsWithProduct_0x1");
}

return uClassPointer;
};

bool __Profile_TA__ValidateLoadoutsWithProduct_0x1(class ULoadoutSet_TA* L);
};

// Class TAGame.LoadoutSet_TA
// 0x0068 (0x0060 - 0x00C8)
class ULoadoutSet_TA : public UObject
{
public:
class FString LoadoutSetName; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class ULoadout_TA* Loadouts[0x2]; // 0x0070 (0x0010)
[0x0000008000000000]
struct FScriptDelegate __EventChanged__Delegate; // 0x0080
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventProductEquipped__Delegate; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventOnlineProductEquipped__Delegate; // 0x00B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutSet_TA");
}

return uClassPointer;
};

void __LoadoutSet_TA__AddListeners_0x3(struct FProductInstanceID InstanceID);
void __LoadoutSet_TA__AddListeners_0x2(int32_t ProductID);
void __LoadoutSet_TA__AddListeners_0x1(class ULoadout_TA* Loadout);
void EnsureConsistentBodies();
void Validate();
bool HasOnlineProduct(struct FProductInstanceID InstanceID);
class UProductAsset_Body_TA* LoadBodyAsset();
int32_t GetBodyProductID();
bool HasProducts();
bool HasProduct(int32_t ProductID);
void AddListeners();
void CopyFromLoadoutSet(class ULoadoutSet_TA* LoadoutSetToCopy);
void InitLoadouts(class UProfile_TA* InProfile);

```

```
static class ULoadoutSet_TA* CreateEmptyLoadoutSet(class UObject* InOuter, class FString InName);
static class ULoadoutSet_TA* CreateLoadoutSet(class UProfile_TA* InProfile, class FString InName);
void EventOnlineProductEquipped(class ULoadoutSet_TA* LoadoutSet, struct FProductInstanceID InstanceID);
void EventProductEquipped(class ULoadoutSet_TA* LoadoutSet, int32_t ProductID);
void EventChanged(class ULoadoutSet_TA* LoadoutSet, class ULoadout_TA* Loadout);
};

// Class TAGame.__RecentPlayers_TA__AddMatchMate_0x1
// 0x0008 (0x0060 - 0x0068)
class U__RecentPlayers_TA__AddMatchMate_0x1 : public UObject
{
public:
    class UPersona_TA* Persona; // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__RecentPlayers_TA__AddMatchMate_0x1");
        }

        return uClassPointer;
    }

    bool __RecentPlayers_TA__AddMatchMate_0x1(struct FPlayerGameID G);
};

// Class TAGame.__RecentPlayers_TA__AddPlayerInternal_0x1
// 0x0008 (0x0060 - 0x0068)
class U__RecentPlayers_TA__AddPlayerInternal_0x1 : public UObject
{
public:
    class UPersona_TA* Persona; // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__RecentPlayers_TA__AddPlayerInternal_0x1");
        }

        return uClassPointer;
    }
}
```

```

};

bool __RecentPlayers_TA__AddPlayerInternal_0x1(struct FPlayerLoadoutData L);
};

// Class TAGame.__RecentPlayers_TA__FilterFriends_0x1
// 0x0008 (0x0060 - 0x0068)
class U__RecentPlayers_TA__FilterFriends_0x1 : public UObject
{
public:
class UPersonas_TA*           PersonasData;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__RecentPlayers_TA__FilterFriends_0x1");
}

return uClassPointer;
};

bool __RecentPlayers_TA__FilterFriends_0x1(struct FPersonaDataId RecentPlayer);
};

// Class TAGame.__RecentPlayersMet_TA__RecordRecentPlayers_0x2
// 0x0038 (0x0060 - 0x0098)
class U__RecentPlayersMet_TA__RecordRecentPlayers_0x2 : public UObject
{
public:
TArray<struct FUniqueNetId>      PlayerIds;          // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FFriendHistoryKey>   Keys;              // 0x0070 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UOnlineSubsystem*           Online;             // 0x0080 (0x0008)
[0x0000000000000000]
class FString                   Presence;           // 0x0088 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__RecentPlayersMet_TA__RecordRecentPlayers_0x2");
}
}

```

```

return uClassPointer;
};

struct FFriendHistoryKey
{
    __RecentPlayersMet_TA__RecordRecentPlayers_0x2 __RecentPlayersMet_TA__RecordRecentPlayers_0x1_0x2(int32_t Index);
    int32_t
    __RecentPlayersMet_TA__RecordRecentPlayers_0x2 __RecentPlayersMet_TA__RecordRecentPlayers_0x1_0x1(struct FUniqueNetId Id);
    void __RecentPlayersMet_TA__RecordRecentPlayers_0x1(TArray<struct FUniqueNetId> AllowedIds);
    struct FFriendHistoryKey __RecentPlayersMet_TA__RecordRecentPlayers_0x3(int32_t Index);
    int32_t __RecentPlayersMet_TA__RecordRecentPlayers_0x2(struct FUniqueNetId Id);
};

// Class TAGame.RecentPlayersMet_TA
// 0x0034 (0x0060 - 0x0094)
class URecentPlayersMet_TA : public UObject
{
public:
    class FString PendingPresence; // 0x0060 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FUniqueNetId> PendingRecentPlayers; // 0x0070
    (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FFriendHistoryKey> PendingRecentPlayerKeys; // 0x0080
    (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t WaitingForPermissionsCount; // 0x0090 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RecentPlayersMet_TA");
        }
    }

    return uClassPointer;
};

void RecordRecentPlayers();
};

// Class TAGame._ReplayManager_TA__PlayReplayFile_0x1
// 0x0020 (0x0060 - 0x0080)
class U_ReplayManager_TA__PlayReplayFile_0x1 : public UObject
{
public:
    class FString Map; // 0x0060 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class FString URL; // 0x0070 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
};

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__ReplayManager_TA__PlayReplayFile_0x1");

}

return uClassPointer;
};

void __ReplayManager_TA__PlayReplayFile_0x1(class UReplayManager_TA* _, class UReplay_TA* Replay, class UError* Error);
};

// Class TAGame.__SaveData_TA__FilterEquippedOnlineProducts_0x1
// 0x0010 (0x0060 - 0x0070)
class U__SaveData_TA__FilterEquippedOnlineProducts_0x1 : public UObject
{
public:
TArray<struct FProductInstanceId> EquippedInstanceIDs; // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__SaveData_TA__FilterEquippedOnlineProducts_0x1");
}

return uClassPointer;
};

bool __SaveData_TA__FilterEquippedOnlineProducts_0x1(class UOnlineProduct_TA* Op);
};

// Class TAGame.__SaveData_TA__HandleOnlineProductsReceived_0x1
// 0x0008 (0x0060 - 0x0068)
class U__SaveData_TA__HandleOnlineProductsReceived_0x1 : public UObject
{
public:
class UProductsSave_TA* ProductsSave; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._SaveData_TA__HandleOnlineProductsReceived_0x1");
}

return uClassPointer;
};

void __SaveData_TA__HandleOnlineProductsReceived_0x1();

// Class TAGame._SaveData_TA__ValidateLoadoutsWithProduct_0x1
// 0x0010 (0x0060 - 0x0070)
class U__SaveData_TA__ValidateLoadoutsWithProduct_0x1 : public UObject
{
public:
struct FProductInstanceID           OnlinID;                      // 0x0060 (0x0010)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._SaveData_TA__ValidateLoadoutsWithProduct_0x1");
}

return uClassPointer;
};

void __SaveData_TA__ValidateLoadoutsWithProduct_0x1(class UProfile_TA* P);
};

// Class TAGame._SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x1
// 0x001C (0x0060 - 0x007C)
class U__SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x1 : public UObject
{
public:
class UCarMeshComponent_TA*          Mesh;                      // 0x0060 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UProductPaint_TA*             Paint;                     // 0x0068 (0x0008)
[0x0000000000000000]
class UAssetAttribute_TeamEdition_TA* TeamEdition;            // 0x0070
(0x0008) [0x0000000000000000]
int32_t                             TeamID;                   // 0x0078 (0x0004)
[0x0000000000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x1");
}

return uClassPointer;
};

void __SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x5(struct FAttachment A);
void __SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x4(struct FWheelFX T);
void __SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x3(struct FFXAttachment A);
void __SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x2(struct
FProductAttachment A);
void __SeqAct_SetLoadout_TA__ApplyNewChangesToProductAssets_0x1(class
UAntennaComponent_TA* A);
};

// Class TAGame.ShopAdsConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UShopAdsConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FShopAdData>           Ads;                      // 0x0078 (0x0010)
[0x00010000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopAdsConfig_TA");
}

return uClassPointer;
};

void HandleImageLoaded(class FString URL, class UTexture2DDynamic* Texture);
void Apply();
};

// Class TAGame._ShopAdsConfig_TA__Apply_0x1
// 0x0068 (0x0060 - 0x00C8)
class U__ShopAdsConfig_TA__Apply_0x1 : public UObject
{
public:
struct FShopAdData           Data;                      // 0x0060 (0x0068)

```

[0x00010000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.__ShopAdsConfig_TA__Apply_0x1");  
}  
  
return uClassPointer;  
};  
  
void __ShopAdsConfig_TA__Apply_0x3(class UTexture2DDynamic* Texture);  
void __ShopAdsConfig_TA__Apply_0x2(class UTexture2DDynamic* Texture);  
void __ShopAdsConfig_TA__Apply_0x1(class UTexture2DDynamic* Texture);  
};  
  
// Class TAGame.GFxData_MenuStack_TA  
// 0x0048 (0x0098 - 0x00E0)  
class UGFxData_MenuStack_TA : public UGFxDataSingleton_X  
{  
public:  
TArray<struct FName> MenuNames; // 0x0098 (0x0010)  
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)  
struct FName LastButtonTriggered; // 0x00A8 (0x0008)  
[0x0000000000002000] (CPF_Transient)  
struct FScriptDelegate __EventTopMenuChanged__Delegate; // 0x00B0  
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)  
struct FScriptDelegate __EventMenuItemTriggered__Delegate; // 0x00C8  
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MenuStack_TA");  
}  
  
return uClassPointer;  
};  
  
void HandleMapLoad(class FString MapName);  
void ButtonTriggered(struct FName ButtonName);  
struct FName GetTopMenu();  
void PopMenu(struct FName MenuName, unsigned long bClearingStack);  
void PushMenu(struct FName MenuName);  
void eventOnRemoved();  
void eventOnShellSet();
```

```

void EventMenuButtonTriggered(struct FName ButtonPress);
void EventTopMenuChanged(struct FName PrevMenu, struct FName TopMenu, unsigned long
bClearingStack);
};

// Class TAGame.NotificationSave_TA
// 0x0068 (0x00C8 - 0x0130)
class UNotificationSave_TA : public USaveObject_TA
{
public:
TArray<class UNotification_TA*> Notifications; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bShowInGameNotifications : 1; // 0x00D8 (0x0004)
[0x0008000000000000] [0x00000001]
unsigned long bShowItemShopNotifications : 1; // 0x00D8 (0x0004)
[0x0008000000000000] [0x00000002]
unsigned long bFirstTimeDefaultShopNotifications : 1; // 0x00D8
(0x0004) [0x0000000000000000] [0x00000004]
unsigned long bShopNotificationsDefaultValue : 1; // 0x00D8
(0x0004) [0x0000000000000000] [0x00000008]
struct FEngagementEventNotificationData EngagementEventNotification[0x2]; // 0x00E0 (0x0020) [0x0000000000000000]
struct FScriptDelegate __bShowInGameNotifications__ChangeNotify; // 0x0100 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __bShowItemShopNotifications__ChangeNotify; // 0x0118 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NotificationSave_TA");
}

return uClassPointer;
};

void __bShowItemShopNotifications__ChangeNotifyFunc();
void __bShowInGameNotifications__ChangeNotifyFunc();
class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
void OnLoad();
void AddNotification(class UNotification_TA* Notification);
};

// Class TAGame.ItemShopNotificationsConfig_TA
// 0x0028 (0x0078 - 0x00A0)
class UItemShopNotificationsConfig_TA : public UOnlineConfig_X
{
public:
unsigned long bItemShopNotificationsEnabled : 1; // 0x0078
(0x0004) [0x0008000040000001] [0x00000001] (CPF_Edit | CPF_EditInlineNotify)

```

```

float PopUpDuration; // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t MinRequiredLevel; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FScriptDelegate __bItemShopNotificationsEnabled__ChangeNotify; //
0x0088 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemShopNotificationsConfig_TA");
}

return uClassPointer;
};

void __bItemShopNotificationsEnabled__ChangeNotifyFunc();

// Class TAGame.ShopsManager_TA
// 0x00C0D8 (0x0060 - 0x012038)
class UShopsManager_TA : public UObject
{
public:
class UShopsService_TA* ServiceArchetype; // 0x0060 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UShopsService_TA* Service; // 0x0068 (0x0008)
[0x0001000000000001] (CPF_Edit)
int32_t ActiveShopID; // 0x0070 (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t ActiveCatalogID; // 0x0074 (0x0004)
[0x0001000000002000] (CPF_Transient)
struct FScriptDelegate __EventShopItemTextureLoaded__Delegate; // 0x0078
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnGetAllShopsResult__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnGetCatalogueResult__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnGetCataloguesResult__Delegate; // 0x00C0
(0x0018) [0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnGetCrateShopResult__Delegate; // 0x00C0D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnPurchaseResultSuccess__Delegate; //
0x00D8F0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnGetItemShopNotificationsResult__Delegate; //
0x00F0108 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OnFailed__Delegate; // 0x01208 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopsManager_TA");
}

return uClassPointer;
};

class ULocalPlayer_X* GetPlayer();
class UASyncTask* GetItemShopNotifications(struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
void HandlePurchaseSuccess(TArray<struct FOnlineProductData> PurchasedProducts,
TArray<struct FCurrency> PurchasedCurrencies);
class UASyncTask* Purchase(int32_t CostID, int32_t ShopID, int32_t DiscountID, int32_t Count,
int32_t ExpectedPriceID, int32_t ExpectedPriceAmount, int32_t ExpectedProductCount, struct
FScriptDelegate SuccessCallback, struct FScriptDelegate FailedCallback);
class UASyncTask* GetCrateShop(int32_t CrateID, struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
void GetShopCatalogues(TArray<int32_t> ShopIDs, struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
class UASyncTask* GetShopCatalogue(int32_t ShopID, struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
class UASyncTask* GetAllShops(struct FScriptDelegate Callback, struct FScriptDelegate
FailedCallback);
void InitServiceProvider();
void Init();
void OnFailed(class UError* Error);
void OnGetItemShopNotificationsResult(TArray<struct FItemShopNotificationData>
ShopNotifications);
void OnPurchaseResultSuccess(TArray<struct FOnlineProductData> PurchasedProducts,
TArray<struct FCurrency> PurchasedCurrency);
void OnGetCrateShopResult(TArray<struct FShopItem> Catalogue, int32_t ShopID);
void OnGetCataloguesResult(TArray<struct FShopCatalogue>& Catalogues);
void OnGetCatalogueResult(TArray<struct FShopItem> Catalogue);
void OnGetAllShopsResult(TArray<struct FShopData> Data);
void EventShopItemTextureLoaded(int32_t ShopID, int32_t ShopItemIndex, class UTexture*  

LoadedTexture);
};

// Class TAGame.ItemShopNotification_TA
// 0x0014 (0x0170 - 0x0184)
class UItemShopNotification_TA : public UNotification_TA
{
public:
class UItemShopNotificationsConfig_TA* ItemShopNotificationsConfig; //  

0x0170 (0x0008) [0x0000800000002000] (CPF_Transient)
class UTexture* Icon; // 0x0178 (0x0008)  

[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
int32_t ShopNotificationID; // 0x0180 (0x0004)  

[0x0000000000000000]

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemShopNotification_TA");
}

return uClassPointer;
};

class UItemShopNotification_TA* SetShopNotificationID(int32_t InShopNotificationID);
class UItemShopNotification_TA* Setup(class UTexture* InIcon);
};

// Class TAGame.__ShopNotificationsManager_TA__LoadNotificationIcon_0x1
// 0x0060 (0x0060 - 0x00C0)
class U__ShopNotificationsManager_TA__LoadNotificationIcon_0x1 : public UObject
{
public:
struct FScriptDelegate           Callback;                      // 0x0060 (0x0018)
[0x00010000000400000] (CPF_NeedCtorLink)
struct FItemShopNotificationData ShopNotificationData;          // 0x0078
(0x0048) [0x00010000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ShopNotificationsManager_TA__LoadNotificationIcon_0x1");
}

return uClassPointer;
};

void __ShopNotificationsManager_TA__LoadNotificationIcon_0x2(struct FAssetLoadResult
AssetResult);
void __ShopNotificationsManager_TA__LoadNotificationIcon_0x1(class UTexture2DDynamic*
LoadedTexture);
};

// Class TAGame.__ShopNotificationsManager_TA__HandleAssetLoaded_0x1
// 0x0060 (0x0060 - 0x00C0)
class U__ShopNotificationsManager_TA__HandleAssetLoaded_0x1 : public UObject
{
public:
struct FScriptDelegate           Callback;                      // 0x0060 (0x0018)
```

```
[0x00010000000400000] (CPF_NeedCtorLink)
struct FItemShopNotificationData           ShopNotificationData;          // 0x0078
(0x0048) [0x00010000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._ShopNotificationsManager_TA_HandleAssetLoaded_0x1");
}

return uClassPointer;
};

void __ShopNotificationsManager_TA_HandleAssetLoaded_0x2(struct FAssetLoadResult
AssetResult);
void __ShopNotificationsManager_TA_HandleAssetLoaded_0x1(struct
FProductThumbnailResult ThumbnailResult);
};

// Class TAGame.ShopMetrics_TA
// 0x0040 (0x0080 - 0x00C0)
class UShopMetrics_TA : public UMetricsGroup_X
{
public:
TArray<struct FShopTabData>           CurrentShopTabMetrics;          // 0x0080
(0x0010) [0x00010000000400000] (CPF_NeedCtorLink)
int32_t                                CurrentBrowseSourceID;          // 0x0090 (0x0004)
[0x0001000000000000]
class UShopNotificationsManager_TA*      ShopNotificationsManager;        //
0x0098 (0x0008) [0x00010800000000000]
TArray<int32_t>                         CurrentShopNotificationIDs;    // 0x00A0 (0x0010)
[0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>                         PastShopNotificationIDs;     // 0x00B0 (0x0010)
[0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopMetrics_TA");
}

return uClassPointer;
};

void ShopNotificationShown(int32_t ShopNotificationID);
```

```
struct FEOSShopPurchaseEventProduct
ShopPurchaseEventProductFromOnlineProductData(struct FOnlineProductData Data);
void MtxPurchase(class FString CurrencyType, int32_t CurrencyAmount, TArray<struct FOnlineProductData> PurchasedProducts);
void ShopPurchase(int32_t CurrencyID, int32_t CurrencyAmount, TArray<struct FOnlineProductData> PurchasedProducts);
void PrintShopMetricsData();
void ShopSessionEnded(int32_t BrowseSourceID, TArray<struct FShopTabData> ShopTabMetrics, TArray<int32_t> ShopNotificationIDs);
void ShopSessionStarted();
void PreviewedShopItem(int32_t ShopItemID);
void IDChanged(int32_t FromShopID, int32_t NewShopID, int32_t FromCatalogID, int32_t NewCatalogID);
};

// Class TAGame.ShopsService_TA
// 0x0000 (0x0060 - 0x0060)
class UShopsService_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopsService_TA");
}

return uClassPointer;
};

class UAsyncTask* GetItemShopNotifications(struct FScriptDelegate Callback, struct FScriptDelegate FailedCallback);
class UAsyncTask* Purchase(struct FUniqueNetId PlayerID, int32_t CostID, int32_t ShopID, int32_t DiscountID, int32_t Count, int32_t ExpectedPriceID, int32_t ExpectedPriceAmount, int32_t ExpectedProductCount, struct FScriptDelegate SuccessCallback, struct FScriptDelegate FailedCallback);
class UAsyncTask* GetCrateShop(int32_t CrateID, struct FScriptDelegate Callback, struct FScriptDelegate FailedCallback);
void GetShopCatalogues(TArray<int32_t> ShopIDs, struct FScriptDelegate Callback, struct FScriptDelegate FailedCallback);
class UAsyncTask* GetShopCatalogue(int32_t ShopID, struct FScriptDelegate Callback, struct FScriptDelegate FailedCallback);
class UAsyncTask* GetAllShops(struct FScriptDelegate Callback, struct FScriptDelegate FailedCallback);
};

// Class TAGame.__ShopsManager_TA__Purchase_0x1
// 0x0030 (0x0060 - 0x0090)
class U_ShopsManager_TA__Purchase_0x1 : public UObject
{
```

```
public:
struct FScriptDelegate           SuccessCallback;          // 0x0060 (0x0018)
[0x00010000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           FailedCallback;         // 0x0078 (0x0018)
[0x00010000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__ShopsManager_TA__Purchase_0x1");
}

return uClassPointer;
};

void __ShopsManager_TA__Purchase_0x2(class UError* Error);
void __ShopsManager_TA__Purchase_0x1(TArray<struct FOnlineProductData>
PurchasedProducts, TArray<struct FCurrency> PurchasedCurrencies);
};

// Class TAGame.__ShopsService_Local_TA__GetAllShops_0x1
// 0x0028 (0x0060 - 0x0088)
class U_ShopsService_Local_TA__GetAllShops_0x1 : public UObject
{
public:
struct FScriptDelegate           Callback;               // 0x0060 (0x0018)
[0x00010000000400000] (CPF_NeedCtorLink)
TArray<struct FShopData>        Shops;                 // 0x0078 (0x0010)
[0x00010000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_Local_TA__GetAllShops_0x1");
}

return uClassPointer;
};

void __ShopsService_Local_TA__GetAllShops_0x1();
};

// Class TAGame.ShopsService_Local_TA
// 0x0004 (0x0060 - 0x0064)
class UShopsService_Local_TA : public UShopsService_TA
```

```
{  
public:  
    float ServiceDelay; // 0x0060 (0x0004)  
    [0x0001000000000001] (CPF_Edit)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.ShopsService_Local_TA");  
        }  
  
        return uClassPointer;  
    };  
  
    class UAsyncTask* QueueTask(class UError* Error);  
    class UAsyncTask* GetItemShopNotifications(struct FScriptDelegate Callback, struct  
        FScriptDelegate FailedCallback);  
    class UAsyncTask* Purchase(struct FUniqueNetId PlayerID, int32_t CostID, int32_t ShopID,  
        int32_t DiscountID, int32_t Count, struct FScriptDelegate SuccessCallback, struct  
        FScriptDelegate FailedCallback);  
    class UAsyncTask* GetCrateShop(int32_t CrateID, struct FScriptDelegate Callback, struct  
        FScriptDelegate FailedCallback);  
    class UAsyncTask* GetShopCatalogue(int32_t ShopID, struct FScriptDelegate Callback, struct  
        FScriptDelegate FailedCallback);  
    class UAsyncTask* GetAllShops(struct FScriptDelegate Callback, struct FScriptDelegate  
        FailedCallback);  
};  
  
// Class TAGame.__ShopsService_Local_TA__GetCrateShop_0x1  
// 0x0028 (0x0060 - 0x0088)  
class U_ShopsService_Local_TA__GetCrateShop_0x1 : public UObject  
{  
public:  
    struct FScriptDelegate Callback; // 0x0060 (0x0018)  
    [0x0001000000400000] (CPF_NeedCtorLink)  
    TArray<struct FShopItem> Items; // 0x0078 (0x0010)  
    [0x0001000000400000] (CPF_NeedCtorLink)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class  
                TAGame.__ShopsService_Local_TA__GetCrateShop_0x1");  
        }  
  
        return uClassPointer;  
    };
```

```
};

void __ShopsService_Local_TA__GetCreateShop_0x1();
};

// Class TAGame.__ShopsService_Local_TA__GetItemShopNotifications_0x1
// 0x0028 (0x0060 - 0x0088)
class U_ShopsService_Local_TA__GetItemShopNotifications_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x0001000000400000] (CPF_NeedCtorLink)
    TArray<struct FItemShopNotificationData> Notifications; // 0x0078
    (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_Local_TA__GetItemShopNotifications_0x1");
        }

        return uClassPointer;
    }

void __ShopsService_Local_TA__GetItemShopNotifications_0x1();
};

// Class TAGame.__ShopsService_Local_TA__GetShopCatalogue_0x1
// 0x0028 (0x0060 - 0x0088)
class U_ShopsService_Local_TA__GetShopCatalogue_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x0001000000400000] (CPF_NeedCtorLink)
    TArray<struct FShopItem>         Items;             // 0x0078 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_Local_TA__GetShopCatalogue_0x1");
        }

        return uClassPointer;
    }
```

```
};

void __ShopsService_Local_TA__GetShopCatalogue_0x1();
};

// Class TAGame.__ShopsService_Local_TA__Purchase_0x1
// 0x0018 (0x0060 - 0x0078)
class U_ShopsService_Local_TA__Purchase_0x1 : public UObject
{
public:
    struct FScriptDelegate           SuccessCallback;           // 0x0060 (0x0018)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__ShopsService_Local_TA__Purchase_0x1");
        }

        return uClassPointer;
    }

void __ShopsService_Local_TA__Purchase_0x1();

};

// Class TAGame.__ShopsService_Local_TA__QueueTask_0x1
// 0x0010 (0x0060 - 0x0070)
class U_ShopsService_Local_TA__QueueTask_0x1 : public UObject
{
public:
    class UAsyncTask*               Task;                      // 0x0060 (0x0008)
    [0x0001000000000000]
    class UError*                  Error;                     // 0x0068 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_Local_TA__QueueTask_0x1");
        }

        return uClassPointer;
    }

void __ShopsService_Local_TA__QueueTask_0x1();
```

```
};

// Class TAGame.__ShopsService_PsyNet_TA__GetAllShops_0x1
// 0x0030 (0x0060 - 0x0090)
class U__ShopsService_PsyNet_TA__GetAllShops_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x0001000000040000] (CPF_NeedCtorLink)
    struct FScriptDelegate           FailedCallback;   // 0x0078 (0x0018)
    [0x0001000000040000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_PsyNet_TA__GetAllShops_0x1");
        }

        return uClassPointer;
    }

    void __ShopsService_PsyNet_TA__GetAllShops_0x2(class URPC_X* RPC);
    void __ShopsService_PsyNet_TA__GetAllShops_0x1(class URPC_GetAllShops_TA* RPC);
};

// Class TAGame.RPC_GetAllShops_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_GetAllShops_TA : public URPC_X
{
public:
    TArray<struct FShopData>           Shops;           // 0x00E8 (0x0010)
    [0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_GetAllShops_TA");
        }

        return uClassPointer;
    }

};

// Class TAGame.ShopsService_PsyNet_TA
```

```

// 0x0000 (0x0060 - 0x0060)
class UShopsService_PsyNet_TA : public UShopsService_TA
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopsService_PsyNet_TA");
}

return uClassPointer;
};

class UAsyncTask* GetItemShopNotifications(struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
class UAsyncTask* Purchase(struct FUniqueNetId PlayerID, int32_t CostID, int32_t ShopID,
int32_t DiscountID, int32_t Count, int32_t ExpectedPriceID, int32_t ExpectedPriceAmount, int32_t
ExpectedProductCount, struct FScriptDelegate SuccessCallback, struct FScriptDelegate
FailedCallback);
class UAsyncTask* GetCrateShop(int32_t CrateID, struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
void GetShopCatalogues(TArray<int32_t> ShopIDs, struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
class UAsyncTask* GetShopCatalogue(int32_t ShopID, struct FScriptDelegate Callback, struct
FScriptDelegate FailedCallback);
class UAsyncTask* GetAllShops(struct FScriptDelegate Callback, struct FScriptDelegate
FailedCallback);
};

// Class TAGame.__ShopsService_PsyNet_TA__GetCrateShop_0x1
// 0x0030 (0x0060 - 0x0090)
class U_ShopsService_PsyNet_TA__GetCrateShop_0x1 : public UObject
{
public:
struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           FailedCallback;    // 0x0078 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_PsyNet_TA__GetCrateShop_0x1");
}
}

```

```

return uClassPointer;
};

void __ShopsService_PsyNet_TA__GetCrateShop_0x2(class URPC_X* RPC);
void __ShopsService_PsyNet_TA__GetCrateShop_0x1(class URPC_GetCrateShop_TA* RPC);
};

// Class TAGame.RPC_GetCrateShop_TA
// 0x001C (0x00E8 - 0x0104)
class URPC_GetCrateShop_TA : public URPC_X
{
public:
int32_t CrateID; // 0x00E8 (0x0004)
[0x0001000000000000]
TArray<struct FShopItem> ShopItems; // 0x00F0 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t ShopID; // 0x0100 (0x0004)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetCrateShop_TA");
}

return uClassPointer;
};

class URPC_GetCrateShop_TA* SetShopID(int32_t InCrateID);
};

// Class TAGame.__ShopsService_PsyNet_TA__GetItemShopNotifications_0x1
// 0x0030 (0x0060 - 0x0090)
class U__ShopsService_PsyNet_TA__GetItemShopNotifications_0x1 : public UObject
{
public:
struct FScriptDelegate Callback; // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate FailedCallback; // 0x0078 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class

```

```

TAGame.__ShopsService_PsyNet_TA__GetItemShopNotifications_0x1");
}

return uClassPointer;
};

void __ShopsService_PsyNet_TA__GetItemShopNotifications_0x2(class URPC_X* RPC);
void __ShopsService_PsyNet_TA__GetItemShopNotifications_0x1(class
URPC_GetItemShopNotifications_TA* RPC);
};

// Class TAGame.RPC_GetItemShopNotifications_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_GetItemShopNotifications_TA : public URPC_X
{
public:
TArray<struct FItemShopNotificationData>      ShopNotifications;           // 0x00E8
(0x0010) [0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetItemShopNotifications_TA");
}

return uClassPointer;
};

};

// Class TAGame.__ShopsService_PsyNet_TA__GetShopCatalogue_0x1
// 0x0030 (0x0060 - 0x0090)
class U_ShopsService_PsyNet_TA__GetShopCatalogue_0x1 : public UObject
{
public:
struct FScriptDelegate          Callback;           // 0x0060 (0x0018)
[0x00010000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          FailedCallback;     // 0x0078 (0x0018)
[0x00010000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__ShopsService_PsyNet_TA__GetShopCatalogue_0x1");
}

```

```

return uClassPointer;
};

void __ShopsService_PsyNet_TA__GetShopCatalogue_0x4(class URPC_X* RPC);
void __ShopsService_PsyNet_TA__GetShopCatalogue_0x3(class URPC_GetShopCatalogue_TA*
RPC);
void __ShopsService_PsyNet_TA__GetShopCatalogue_0x2(class URPC_X* RPC);
void __ShopsService_PsyNet_TA__GetShopCatalogue_0x1(class
URPC_GetShopCatalogueV2_TA* RPC);
};

// Class TAGame.RPC_GetShopCatalogueV2_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_GetShopCatalogueV2_TA : public URPC_X
{
public:
TArray<int32_t> ShopIDs; // 0x00E8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FShopCatalogue> Catalogues; // 0x00F8 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetShopCatalogueV2_TA");
}

return uClassPointer;
};

class URPC_GetShopCatalogueV2_TA* SetShopIDs(TArray<int32_t> InShopIDs);

// Class TAGame.RPC_GetShopCatalogue_TA
// 0x0018 (0x00E8 - 0x0100)
class URPC_GetShopCatalogue_TA : public URPC_X
{
public:
int32_t ShopID; // 0x00E8 (0x0004)
[0x0001000000000000]
TArray<struct FShopItem> ShopItems; // 0x00F0 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetShopCatalogue_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_GetShopCatalogue_TA* SetShopID(int32_t InShopID);  
};  
  
// Class TAGame.__ShopsService_PsyNet_TA__GetShopCatalogues_0x1  
// 0x0030 (0x0060 - 0x0090)  
class U_ShopsService_PsyNet_TA__GetShopCatalogues_0x1 : public UObject  
{  
public:  
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)  
    [0x0000000000400000] (CPF_NeedCtorLink)  
    struct FScriptDelegate           FailedCallback;     // 0x0078 (0x0018)  
    [0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class  
TAGame.__ShopsService_PsyNet_TA__GetShopCatalogues_0x1");  
        }  
  
        return uClassPointer;  
    };  
  
void __ShopsService_PsyNet_TA__GetShopCatalogues_0x2(class URPC_X* RPC);  
void __ShopsService_PsyNet_TA__GetShopCatalogues_0x1(class  
URPC_GetShopCatalogueV2_TA* RPC);  
};  
  
// Class TAGame.__ShopsService_PsyNet_TA__Purchase_0x1  
// 0x0030 (0x0060 - 0x0090)  
class U_ShopsService_PsyNet_TA__Purchase_0x1 : public UObject  
{  
public:  
    struct FScriptDelegate           SuccessCallback;     // 0x0060 (0x0018)  
    [0x0001000000400000] (CPF_NeedCtorLink)  
    struct FScriptDelegate           FailedCallback;      // 0x0078 (0x0018)  
    [0x0001000000400000] (CPF_NeedCtorLink)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._ShopsService_PsyNet_TA__Purchase_0x1");
}

return uClassPointer;
};

void __ShopsService_PsyNet_TA__Purchase_0x4(class URPC_X* RPC);
void __ShopsService_PsyNet_TA__Purchase_0x3(class URPC_PurchaseItemFromShop_TA*
RPC);
void __ShopsService_PsyNet_TA__Purchase_0x2(class URPC_X* RPC);
void __ShopsService_PsyNet_TA__Purchase_0x1(class URPC_PurchaseItemFromShopV2_TA*
RPC);
};

// Class TAGame.RPC_PurchaseItemFromShop_TA
// 0x0080 (0x00E8 - 0x0168)
class URPC_PurchaseItemFromShop_TA : public URPC_X
{
public:
int32_t ShopItemCostID; // 0x00E8 (0x0004)
[0x0001000000000000]
int32_t ShopID; // 0x00EC (0x0004)
[0x0001000000000000]
int32_t ShopDiscountID; // 0x00F0 (0x0004)
[0x0001000000000000]
struct FUniqueNetId PlayerID; // 0x00F8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t Count; // 0x0140 (0x0004)
[0x0001000000000000]
TArray<struct FOnlineProductData> ProductData; // 0x0148
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCurrency> CurrencyData; // 0x0158 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_PurchaseItemFromShop_TA");
}

return uClassPointer;
};

class URPC_PurchaseItemFromShop_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
class URPC_PurchaseItemFromShop_TA* SetCount(int32_t InCount);
class URPC_PurchaseItemFromShop_TA* SetShopID(int32_t InShopID);
class URPC_PurchaseItemFromShop_TA* SetCostID(int32_t InCostID, int32_t InDiscountID);

```

```
};

// Class TAGame.RPC_PurchaseItemFromShopV2_TA
// 0x0014 (0x0168 - 0x017C)
class URPC_PurchaseItemFromShopV2_TA : public URPC_PurchaseItemFromShop_TA
{
public:
TArray<struct FShopPrice>           ExpectedPrice;           // 0x0168 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t                           ExpectedProductCount;    // 0x0178 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_PurchaseItemFromShopV2_TA");
}

return uClassPointer;
};

class URPC_PurchaseItemFromShopV2_TA* SetExpectedProductCount(int32_t
InExpectedProductCount);
class URPC_PurchaseItemFromShopV2_TA* SetExpectedPrice(int32_t ExpectedPriceID, int32_t
ExpectedPriceAmount);
};

// Class TAGame.__SpecialEventConfig_TA__SyncImageForIndex_0x1
// 0x0004 (0x0060 - 0x0064)
class U__SpecialEventConfig_TA__SyncImageForIndex_0x1 : public UObject
{
public:
int32_t                           I;                      // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__SpecialEventConfig_TA__SyncImageForIndex_0x1");
}

return uClassPointer;
};

void __SpecialEventConfig_TA__SyncImageForIndex_0x1(class UTexture2DDynamic* Texture);
```

```

};

// Class TAGame.SpecialEventConfig_TA
// 0x01C8 (0x0078 - 0x0240)
class USpecialEventConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FSpecialEventStoreConfigs>      StoreConfigs;           // 0x0078
(0x0010) [0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FDebugMicroEventPlaylistID>      DebugMicroEventPlaylistIDs; // 0x0088 (0x0010) [0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t          CurrencyID;           // 0x0098 (0x0004)
[0x000800040000001] (CPF>Edit | CPF>EditInlineNotify)
int32_t          ShopID;              // 0x009C (0x0004)
[0x000000040000001] (CPF>Edit | CPF>EditInlineNotify)
class FString    Title;                // 0x00A0 (0x0010)
[0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    Subtitle;             // 0x00B0 (0x0010)
[0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    RedemptionTitle;       // 0x00C0 (0x0010)
[0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    Description;           // 0x00D0 (0x0010)
[0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    RedemptionDescription; // 0x00E0 (0x0010)
[0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    CurrencyName;         // 0x00F0 (0x0010)
[0x000000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    CurrencyImageURL;     // 0x0100 (0x0010)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
class FString    CurrencyImageLargeURL; // 0x0110 (0x0010)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
class FString    BackgroundLogoURL;    // 0x0120 (0x0010)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
class FString    BackgroundImageURL;   // 0x0130 (0x0010)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
class FString    LogolImageURL;        // 0x0140 (0x0010)
[0x000000000400001] (CPF>Edit | CPF_NeedCtorLink)
int32_t          CurrencyColor;       // 0x0150 (0x0004)
[0x000000040000001] (CPF>Edit | CPF>EditInlineNotify)
int32_t          BackgroundColor;     // 0x0154 (0x0004)
[0x000000040000001] (CPF>Edit | CPF>EditInlineNotify)
uint64_t          StartTime;           // 0x0158 (0x0008)
[0x000000040002000] (CPF>Transient | CPF>EditInlineNotify)
class FString    StartTimeString;      // 0x0160 (0x0010)
[0x000000040402000] (CPF>Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
uint64_t          EndTime;             // 0x0170 (0x0008)
[0x000000040002000] (CPF>Transient | CPF>EditInlineNotify)
class FString    EndTimeString;        // 0x0178 (0x0010)
[0x000000040402000] (CPF>Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
uint64_t          RedemptionEndTime;   // 0x0188 (0x0008)
[0x000000040002000] (CPF>Transient | CPF>EditInlineNotify)
class FString    RedemptionEndTimeString; // 0x0190 (0x0010)
[0x000000040402000] (CPF>Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString    EventFolderName;       // 0x01A0 (0x0010)

```

```

[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture*           BackgroundLogo;          // 0x01B0 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture*           BackgroundImage;        // 0x01B8 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture*           CurrencyImage;         // 0x01C0 (0x0008)
[0x0008000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture*           CurrencyImageLarge;    // 0x01C8 (0x0008)
[0x0008000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture*           LogolImage;            // 0x01D0 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
unsigned long              bApplied : 1;          // 0x01D8 (0x0004)
[0x0008000000002000] [0x00000001] (CPF_Transient)
struct FScriptDelegate     __CurrencyID__ChangeNotify; // 0x01E0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __CurrencyImage__ChangeNotify; // 0x01F8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __CurrencyImageLarge__ChangeNotify; // 0x0210
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate     __bApplied__ChangeNotify;      // 0x0228
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialEventConfig_TA");
}

return uClassPointer;
};

void __SpecialEventConfig_TA__Apply_0x5(class UTexture2DDynamic* Texture);
void __SpecialEventConfig_TA__Apply_0x4(class UTexture2DDynamic* Texture);
void __SpecialEventConfig_TA__Apply_0x3(class UTexture2DDynamic* Texture);
void __SpecialEventConfig_TA__Apply_0x2(class UTexture2DDynamic* Texture);
void __SpecialEventConfig_TA__Apply_0x1(class UTexture2DDynamic* Texture);
void __bApplied__ChangeNotifyFunc();
void __CurrencyImageLarge__ChangeNotifyFunc();
void __CurrencyImage__ChangeNotifyFunc();
void __CurrencyID__ChangeNotifyFunc();
bool IsDebugMicroEventPlaylist(int32_t PlaylistId);
int32_t GetSecondsRemaining();
uint8_t GetState();
void SyncImageForIndex(class UWebImageCache_X* WebImageCache, int32_t I);
void Apply();
};

// Class TAGame.__StatusObserver_Products_TA__UpdateTriggersBySlot_0x1
// 0x0004 (0x0060 - 0x0064)
class U__StatusObserver_Products_TA__UpdateTriggersBySlot_0x1 : public UObject

```

```
{  
public:  
int32_t InSlotIndex; // 0x0060 (0x0004)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._StatusObserver_Products_TA__UpdateTriggersBySlot_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __StatusObserver_Products_TA__UpdateTriggersBySlot_0x1(class UStatusTrigger_X* T);  
};  
  
// Class TAGame.StatusTrigger_Product_TA  
// 0x0008 (0x0080 - 0x0088)  
class UStatusTrigger_Product_TA : public UStatusTrigger_X  
{  
public:  
class UProduct_TA* ConditionalValue; // 0x0080 (0x0008)  
[0x0000000000000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_Product_TA");  
}  
  
return uClassPointer;  
};  
  
void DebugPrint();  
void HandleChange(int32_t InProductID);  
};  
  
// Class TAGame.StatusObserver_Products_TA  
// 0x0050 (0x00A0 - 0x00F0)  
class UStatusObserver_Products_TA : public UStatusObserver_X  
{  
public:  
TArray<int32_t> LastEquippedProducts; // 0x00A0 (0x0010)  
[0x0000000000400000] (CPF_NeedCtorLink)
```

```

TArray<int32_t> LastOwnedProducts; // 0x00B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t LastTeamIndex; // 0x00C0 (0x0004)
[0x0000000000000000]
class USaveData_TA* SaveData; // 0x00C8 (0x0008)
[0x0000000000000000]
class UProductsSave_TA* ProductsSave; // 0x00D0 (0x0008)
[0x0000000000000000]
class UProfileLoadoutSave_TA* ProfileLoadoutSave; // 0x00D8
(0x0008) [0x0000000000000000]
TArray<int32_t> PendingChangedProductSlots; // 0x00E0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusObserver_Products_TA");
}

return uClassPointer;
};

void __StatusObserver_Products_TA__Init_0x2(class UPlayerVanitySave_TA* P);
void __StatusObserver_Products_TA__Init_0x1(class UPlayerVanitySave_TA* P);
void __StatusObserver_Products_TA__HandleVanitySave_0x1(struct FProductInstanceID P);
void __StatusObserver_Products_TA__RebuildOwnedProducts_0x3(int32_t SlotIndex);
bool __StatusObserver_Products_TA__RebuildOwnedProducts_0x2(int32_t Id);
int32_t __StatusObserver_Products_TA__RebuildOwnedProducts_0x1(class UOnlineProduct_TA* Op);
void DebugPrint();
void UpdateOwnedTrigger(class UStatusTrigger_ProductOwned_TA* InTrigger);
void HandleProductAdded(class USaveData_TA* InSaveData, class UOnlineProduct_TA* OnlineProduct, class FString Messages);
void HandleProductRemoved(class USaveData_TA* InSaveData, class UOnlineProduct_TA* OnlineProduct);
void HandleOnlineProductsSynced(class USaveData_TA* InSaveData);
void RebuildOwnedProducts();
void HandleProductSave(class UProductsSave_TA* InSaveData);
void UpdateEquippedTrigger(class UStatusTrigger_ProductEquipped_TA* InTrigger);
void OnTeamChanged(int32_t InTeamIndex);
void HandlePreviewTeamChanged(class UProfileLoadoutSave_TA* InLoadoutSave);
void HandleTeamChanged(class APRI_X* InPRI);
void HandlePRIReceived(class APRI_X* InPRI);
void HandleReceivePlayerController(class APlayerController* PlayerRef);
void HandleProductEquipped(int32_t ProductID);
void HandleVanitySave(class UPlayerVanitySave_TA* InVanitySave, int32_t SlotIndex);
void HandleLoadout(class UProfileLoadoutSave_TA* LoadoutSave);
void UpdateTriggersBySlot(class UClass* InType, int32_t InSlotIndex);
void HandleSaveDataLoaded(class USaveData_TA* InSaveData);
void Init(class ULocalPlayer_X* LP);

```

```
};

// Class TAGame.__StatusTriggerManager_TA__AddTriggerCondition_0x1
// 0x0008 (0x0060 - 0x0068)
class U_StatusTriggerManager_TA__AddTriggerCondition_0x1 : public UObject
{
public:
    class UTriggerCondition_TA*           InCondition;          // 0x0060 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__StatusTriggerManager_TA__AddTriggerCondition_0x1");
        }

        return uClassPointer;
    };

    void __StatusTriggerManager_TA__AddTriggerCondition_0x1(class UTriggerClump_TA* G);
};

// Class TAGame.StatusTriggerManager_TA
// 0x0020 (0x0060 - 0x0080)
class UStatusTriggerManager_TA : public UObject
{
public:
    TArray<class UStatusObserver_X*>      Observers;          // 0x0060 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)
    TArray<class UTriggerCondition_TA*>    Conditions;        // 0x0070
    (0x0010) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.StatusTriggerManager_TA");
        }

        return uClassPointer;
    };

    void __StatusTriggerManager_TA__Init_0x4(class UStatusObserver_X* O);
    class UStatusObserver_X* __StatusTriggerManager_TA__Init_0x1(class UClass* Ob);
    void DebugPrint();
    void CheckModifiedCondition(class UTriggerCondition_TA* InCondition);
};
```

```

void RemoveTriggers(class UTriggerCondition_TA* InCondition);
void RemoveTriggerCondition(class UTriggerCondition_TA* InCondition);
void RegisterTriggers(class UTriggerCondition_TA* InCondition, TArray<class UStatusTrigger_X*>
InTriggers);
void AddTriggerCondition(class UTriggerCondition_TA* InCondition);
void Init(class ULocalPlayer_TA* LP);
};

// Class TAGame.__StatusTriggerManager_TA__Init_0x2
// 0x0010 (0x0060 - 0x0070)
class U_StatusTriggerManager_TA__Init_0x2 : public UObject
{
public:
class ULocalPlayer_TA*           LP;                      // 0x0060 (0x0008)
[0x0000000000000000]
class UTriggerCondition_TA*      TriggerCondition;     // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__StatusTriggerManager_TA__Init_0x2");
}

return uClassPointer;
};

void __StatusTriggerManager_TA__Init_0x3(class UTriggerClump_TA* G);
void __StatusTriggerManager_TA__Init_0x2(class UStatusObserver_X* O);
};

// Class TAGame.__StatusTriggerManager_TA__RegisterTriggers_0x1
// 0x0018 (0x0060 - 0x0078)
class U_StatusTriggerManager_TA__RegisterTriggers_0x1 : public UObject
{
public:
class UTriggerCondition_TA*      InCondition;          // 0x0060 (0x0008)
[0x0000000000000000]
TArray<class UStatusTrigger_X*>    InTriggers;        // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StatusTriggerManager_TA__RegisterTriggers_0x1");
}
}

```

```

}

return uClassPointer;
};

void __StatusTriggerManager_TA__RegisterTriggers_0x2(class UStatusObserver_X* O);
void __StatusTriggerManager_TA__RegisterTriggers_0x1();
};

// Class TAGame.__StatusTriggerManager_TA__RemoveTriggers_0x1
// 0x0008 (0x0060 - 0x0068)
class U_StatusTriggerManager_TA__RemoveTriggers_0x1 : public UObject
{
public:
class UTriggerClump_TA*           Clump;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StatusTriggerManager_TA__RemoveTriggers_0x1");
}

return uClassPointer;
};

void __StatusTriggerManager_TA__RemoveTriggers_0x1(class UStatusObserver_X* O);
};

// Class TAGame.__StayAsPartySystem_TA__CreateVoteStartEvent_0x1
// 0x0010 (0x0060 - 0x0070)
class U_StayAsPartySystem_TA__CreateVoteStartEvent_0x1 : public UObject
{
public:
class UServerStartVoteEvent_TA*     StartEvent;      // 0x0060 (0x0008)
[0x0000000000000000]
class ATeam_TA*                   Team;           // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StayAsPartySystem_TA__CreateVoteStartEvent_0x1");
}
}

```

```

return uClassPointer;
};

void __StayAsPartySystem_TA__CreateVoteStartEvent_0x1(class APRI_TA* Leader);
};

// Class TAGame.ServerStartVoteEvent_TA
// 0x0018 (0x0060 - 0x0078)
class UServerStartVoteEvent_TA : public UObject
{
public:
TArray<struct FPartyInfo>          Parties;           // 0x0060 (0x0010)
[0x0000000000040000] (CPF_NeedCtorLink)
class AActor*                      VoteOwner;        // 0x0070 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerStartVoteEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.StayAsPartySystem_TA
// 0x0000 (0x0060 - 0x0060)
class UStayAsPartySystem_TA : public UObject
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StayAsPartySystem_TA");
}

return uClassPointer;
};

static class AStayAsPartyVoter_TA*
__StayAsPartySystem_TA__StartStayAsPartyVote_0x1(struct FPartyInfo Party);
static bool __StayAsPartySystem_TA__ProcessVote_0x1(class AStayAsPartyVoter_TA*

```

```
PartyVoter);
static bool IsOneParty(TArray<class APRI_TA*>& PRIs);
static class UGFxData_PRI_TA* GetPRIData(class AGFxHUD_TA* InHUD, class APRI_TA* InPRI);
static void PlayerSentPartyInvites(class APRI_TA* PRI);
static bool DidPlayerSendPartyInvites(class APRI_TA* PRI);
static TArray<class APRI_TA*> FindHumanTeammatesOfID(TArray<class APRI_TA*> PRIs, struct FUniqueNetId PlayerID);
static bool IsLocalPlayerActive(class ULocalPlayer* LP);
static void StayAsPartyLeader_Removed(class UOnlinePlayer_TA* OnlinePlayer, class UStayAsPartyLeader_TA* Leader);
static void StayAsPartyLeader_Added(class UOnlinePlayer_TA* OnlinePlayer, class UStayAsPartyLeader_TA* Leader);
static void SetVoteActive(TArray<class APRI_TA*> PRIs, unsigned long bActive);
static void GFxPlayerDisconnected(class UNetMode_Client* NetMode, class APRI_TA* PRI, class UPlayerRemovedEvent_TA* DisconnectedEvent);
static void GFxLeftParty(class UNetMode_Client* NetMode, class APRI_TA* PRI, class UPartyLeaderChangedEvent_TA* LeaderChanged);
static void GFxPlayerJoinedParty(class UNetMode_Client* NetMode, class ATeam_TA* Team, class APRI_TA* PRI, class UPartyLeaderChangedEvent_TA* LeaderChanged);
static void GFxLeaderActive(class UNetMode_Client* NetMode, class APRI_TA* PRI, class AStayAsPartyVoter_TA* Voter, class AStayAsPartyVoteYes_TA* YesVote);
static void GFxDisableButton(class UGFxData_GameEvent_TA* GFxGame, class AStayAsPartyVoteYes_TA*_);
static void GFxShowStayAsPartyButton(class UGFxData_GameEvent_TA* GFxGame, class UGFxStayAsPartyDisplayButton_TA* DisplayCommand);
static void FormParty(class APRI_TA* LeaderPRI, class UServerFormPartyEvent_TA* FormEvent);
static void MergeEventConfirmFollower(class UMergePartyEvent_TA* MergeEvent, class ATeam_TA* Team, class APRI_TA* ConfirmedFollower);
static void HandleClientConfirmedMergeInvite(class UMergePartyEvent_TA* MergeEvent, class ATeam_TA* Team, class APRI_TA* ConfirmedFollower, class UServerInviteeConfirmedMergeEvent_TA*_);
static void HandleClientConfirmedVoteInvite(class UStayAsPartyVote_TA* Vote, class ATeam_TA* Team, class APRI_TA* ConfirmedFollower, class UServerInviteeConfirmedMergeEvent_TA*_);
static void BroadcastPartyEvent(class APRI_TA* PRI);
static void UnregisterForLeaderChanged(class APRI_TA* PRI);
static void RegisterForLeaderChanged(class APRI_TA* PRI);
static void UnsubscribeToLeaderChanges(class APRI_TA* PRI);
static void ReportLeaderChanges(class APRI_TA* PRI, class UClientStayAsPartyVoteBeginEvent_TA*_);
static void ReplicateVoteYes_Removed(class UStayAsPartyVote_TA* Vote, class APRI_TA* PRI, class AStayAsPartyVoter_TA* Voter, class AStayAsPartyVoteYes_TA* InVoteYes);
static void ReplicateVoteYes_Added(class UNetMode_Server* NetMode, class APRI_TA* PRI, class AStayAsPartyVoteYes_TA* YesVote);
static void ReplicateVoter_Removed(class UNetMode_Server* NetMode, class APRI_TA* PRI, class AStayAsPartyVoter_TA* PartyVoter);
static void ReplicateVoter_Added(class UNetMode_Server* NetMode, class APRI_TA* PRI, class AStayAsPartyVoter_TA* PartyVoter);
static void VoteYes_Removed(class UStayAsPartyVote_TA* Vote, class AStayAsPartyVoter_TA* Voter, class AStayAsPartyVoteYes_TA* InVoteYes);
static void VoteYes_Added(class AStayAsPartyVoter_TA* Voter, class AStayAsPartyVoteYes_TA* InVoteYes);
static void ReplicateInviteeMergeEvent(class UNetMode_Server* NetMode, class APRI_TA* PRI, class UMergePartyEvent_TA* MergeEvent);
```

```

static void ProcessVote(class UStayAsPartyVote_TA* Vote, class AStayAsPartyVoter_TA* Voter,
class AStayAsPartyVoteYes_TA* PlayerVote);
static void HandlePartyChanged(class UStayAsPartyVote_TA* Vote, class
UPartyLeaderChangedEvent_TA* PartyEvent);
static void VoterRemoved(class UStayAsPartyVote_TA* Vote, class AStayAsPartyVoter_TA*
Voter);
static void VoterDisconnected(class UNetMode_Server* NetMode, class AStayAsPartyVoter_TA*
Voter, class UPlayerRemovedEvent_TA* DisconnectedEvent);
static void AddVoterToVote(class UStayAsPartyVote_TA* Vote, class AStayAsPartyVoter_TA*
Voter);
static void NotifyPlayersVoteStarted(class APlayerController_TA* InController, class
UServerVoteStartedEvent_TA* _);
static void StartStayAsPartyVote(class UServerStartVoteEvent_TA* InEvent);
static void CreateVoteStartEvent(class UNetMode_Server* NetMode, class ATeam_TA* Team,
class UGameEventFinishedState_TA* _);
static void ProcessEndMatch(class UNetMode_Client* NetMode, class ATeam_TA* Team, class
UGameEventFinishedState_TA* _);
};

// Class TAGame.__StayAsPartySystem_TA__FindHumanTeammatesOfID_0x1
// 0x0014 (0x0060 - 0x0074)
class U__StayAsPartySystem_TA__FindHumanTeammatesOfID_0x1 : public UObject
{
public:
TArray<class APRI_TA*> PRIs; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t PlayerIndex; // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StayAsPartySystem_TA__FindHumanTeammatesOfID_0x1");
}

return uClassPointer;
};

bool __StayAsPartySystem_TA__FindHumanTeammatesOfID_0x1(class APRI_TA* P);
};

// Class TAGame.__StayAsPartySystem_TA__GFxLeaderActive_0x1
// 0x0018 (0x0060 - 0x0078)
class U__StayAsPartySystem_TA__GFxLeaderActive_0x1 : public UObject
{
public:
class AStayAsPartyVoter_TA* Voter; // 0x0060 (0x0008)
[0x0000000000000000]
TArray<class APRI_TA*> PrimaryGroup; // 0x0068 (0x0010)

```

[0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._StayAsPartySystem_TA_GFxLeaderActive_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __StayAsPartySystem_TA_GFxLeaderActive_0x2(class APRI_TA* PRI);  
bool __StayAsPartySystem_TA_GFxLeaderActive_0x1(class APRI_TA* PRI);  
};  
  
// Class TAGame._StayAsPartySystem_TA_GFxPlayerJoinedParty_0x1  
// 0x0008 (0x0060 - 0x0068)  
class U_StayAsPartySystem_TA_GFxPlayerJoinedParty_0x1 : public UObject  
{  
public:  
class UPartyLeaderChangedEvent_TA*           LeaderChanged;           // 0x0060  
(0x0008) [0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._StayAsPartySystem_TA_GFxPlayerJoinedParty_0x1");  
}  
  
return uClassPointer;  
};  
  
bool __StayAsPartySystem_TA_GFxPlayerJoinedParty_0x1(class APRI_TA* PRI);  
};  
  
// Class TAGame.PartyLeaderChangedEvent_TA  
// 0x00A8 (0x0060 - 0x0108)  
class UPartyLeaderChangedEvent_TA : public UObject  
{  
public:  
struct FUniqueNetId           PlayerID;           // 0x0060 (0x0048)  
[0x0000000000400000] (CPF_NeedCtorLink)  
struct FUniqueNetId           NewLeaderID;         // 0x00A8 (0x0048)  
[0x0000000000400000] (CPF_NeedCtorLink)
```

```

TArray<struct FUniqueNetId>           NewFollowerIds;          // 0x00F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class AActor*                          VoterOwner;            // 0x0100 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyLeaderChangedEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.__StayAsPartySystem_TA__HandlePartyChanged_0x1
// 0x0008 (0x0060 - 0x0068)
class U__StayAsPartySystem_TA__HandlePartyChanged_0x1 : public UObject
{
public:
class UPartyLeaderChangedEvent_TA*      PartyEvent;           // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StayAsPartySystem_TA__HandlePartyChanged_0x1");
}

return uClassPointer;
};

bool __StayAsPartySystem_TA__HandlePartyChanged_0x1(class AStayAsPartyVoter_TA* Voter);
};

// Class TAGame.__StayAsPartySystem_TA__IsOneParty_0x1
// 0x0010 (0x0060 - 0x0070)
class U__StayAsPartySystem_TA__IsOneParty_0x1 : public UObject
{
public:
TArray<class APRI_TA*>                 PRIs;                // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._StayAsPartySystem_TA_IsOneParty_0x1");
}

return uClassPointer;
};

bool __StayAsPartySystem_TA_IsOneParty_0x1(class APRI_TA* P);
};

// Class TAGame._StayAsPartySystem_TA_MergeEventConfirmFollower_0x1
// 0x0008 (0x0060 - 0x0068)
class U_StayAsPartySystem_TA_MergeEventConfirmFollower_0x1 : public UObject
{
public:
class UMergePartyEvent_TA*           MergeEvent;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._StayAsPartySystem_TA_MergeEventConfirmFollower_0x1");
}

return uClassPointer;
};

bool __StayAsPartySystem_TA_MergeEventConfirmFollower_0x1(class APRI_TA* PRI);
};

// Class TAGame.MergePartyEvent_TA
// 0x0068 (0x0060 - 0x00C8)
class UMergePartyEvent_TA : public UObject
{
public:
struct FUniqueNetId           LeaderID;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId>     InviteIDs;        // 0x00A8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId>     ConfirmedInvitees; // 0x00B8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MergePartyEvent_TA");
}

return uClassPointer;
};

class FString ToTestString();
bool PlayerIsFollower(struct FUniqueNetId& PlayerID);
bool FromReplicatedData(struct FMergePartyReplicatedData& InputData);
void ToReplicatedData(struct FMergePartyReplicatedData& OutputData);
bool AllInviteesConfirmed();
void AddInviteeConfirmation(class APRI_TA* PRI);
bool RemoveInvitee(struct FUniqueNetId& PlayerID);
void AddInvitees(TArray<struct FUniqueNetId>& Ids);
void AddInvitee(struct FUniqueNetId& Id);
class UMergePartyEvent_TA* Copy();
};

// Class TAGame.__StayAsPartySystem_TA__ProcessVote_0x2
// 0x0008 (0x0060 - 0x0068)
class U__StayAsPartySystem_TA__ProcessVote_0x2 : public UObject
{
public:
class AStayAsPartyVoter_TA* VoterIter; // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StayAsPartySystem_TA__ProcessVote_0x2");
}

return uClassPointer;
};

bool __StayAsPartySystem_TA__ProcessVote_0x2(class UMergePartyEvent_TA* Merge);
};

// Class TAGame.__StayAsPartySystem_TA__SetVoteActive_0x1
// 0x0004 (0x0060 - 0x0064)
class U__StayAsPartySystem_TA__SetVoteActive_0x1 : public UObject
{
public:

```

```
unsigned long           bActive : 1;          // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._StayAsPartySystem_TA_SetVoteActive_0x1");

}

return uClassPointer;
};

void __StayAsPartySystem_TA_SetVoteActive_0x1(class APRI_TA* PRI);
};

// Class TAGame._StayAsPartyVote_TA_AddFollowerToLeader_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_StayAsPartyVote_TA_AddFollowerToLeader_0x1 : public UObject
{
public:
struct FUniqueNetId           LeaderID;          // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._StayAsPartyVote_TA_AddFollowerToLeader_0x1");

}

return uClassPointer;
};

bool __StayAsPartyVote_TA_AddFollowerToLeader_0x1(class AStayAsPartyVoter_TA* Voter);
};

// Class TAGame.StayAsPartyVote_TA
// 0x0050 (0x0060 - 0x00B0)
class UStayAsPartyVote_TA : public UObject
{
public:
TArray<class AStayAsPartyVoter_TA*>      Voters;          // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class AStayAsPartyVoter_TA*>      VotedYes;        // 0x0070
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
```

```

TArray<class UMergePartyEvent_TA*> PendingMerges; // 0x0080
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
TArray<class UMergePartyEvent_TA*> MergeRecord; // 0x0090
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPartyUpMergeError> MergeErrors; // 0x00A0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StayAsPartyVote_TA");
}

return uClassPointer;
};

void RemoveVote(class AStayAsPartyVoter_TA* Voter);
bool HasVoted(struct FUniqueNetId& PlayerID);
void ComposePartyChangeWithMerges(class UPartyLeaderChangedEvent_TA* PartyEvent);
void RemovePlayerFromMerges(class AStayAsPartyVoter_TA* RemovedVoter);
void AddPendingMerge(class UMergePartyEvent_TA* MergeEvent);
bool CanPlayerVote(struct FUniqueNetId& PlayerID);
class AStayAsPartyVoter_TA* FindVoter(struct FUniqueNetId& PlayerID);
void AddFollowerToLeader(struct FUniqueNetId& LeaderID, struct FUniqueNetId& FollowerID);
void RemoveFollower(struct FUniqueNetId& FollowerID);
};

// Class TAGame.__StayAsPartyVote_TA__RemoveFollower_0x1
// 0x0048 (0x0060 - 0x00A8)
class U__StayAsPartyVote_TA__RemoveFollower_0x1 : public UObject
{
public:
struct FUniqueNetId FollowerID; // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__StayAsPartyVote_TA__RemoveFollower_0x1");
}

return uClassPointer;
};

void __StayAsPartyVote_TA__RemoveFollower_0x1(class AStayAsPartyVoter_TA* Voter);

```

```

};

// Class TAGame._TAsyncResult_PrivilegeCheck_X_Copy_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TAsyncResult_PrivilegeCheck_X_Copy_0x1 : public UObject
{
public:
class UTAsyncResult_PrivilegeCheck_X*           Instance;          // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TAsyncResult_PrivilegeCheck_X_Copy_0x1");
}

return uClassPointer;
};

void __TAsyncResult_PrivilegeCheck_X_Copy_0x1(class UPrivilegeCheck_X* R, class UError*
E);
};

// Class TAGame.TAsyncResult_PrivilegeCheck_X
// 0x0050 (0x00D0 - 0x0120)
class UTAsyncResult_PrivilegeCheck_X : public UAsyncTask
{
public:
class UPrivilegeCheck_X*           Result;          // 0x00D0 (0x0008)
[0x0000004000000000]
struct FScriptDelegate           __EventResult__Delegate;      // 0x00D8 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventResultComplete__Delegate; // 0x00F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __ResultDelegate__Delegate;    // 0x0108
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TAsyncResult_PrivilegeCheck_X");
}

return uClassPointer;
};

```

```

static class UTAsyncResult__PrivilegeCheck_X* CreateResultError(class UError* InError);
static class UTAsyncResult__PrivilegeCheck_X* CreateResult(class UPrivilegeCheck_X* InResult);
class UTAsyncResult__PrivilegeCheck_X* Copy();
void eventClearCallbacks();
class UTAsyncResult__PrivilegeCheck_X* eventSetResultWhen(class UAsyncTask* Other, struct FScriptDelegate GetResultDelegate);
class UTAsyncResult__PrivilegeCheck_X* eventSetResult(class UPrivilegeCheck_X* InResult, class UError* InError);
class UTAsyncResult__PrivilegeCheck_X* NotifyOnResultComplete(struct FScriptDelegate Callback);
class UTAsyncResult__PrivilegeCheck_X* NotifyOnResult(struct FScriptDelegate Callback);
class UPrivilegeCheck_X* ResultDelegate();
void EventResultComplete(class UPrivilegeCheck_X* OutResult, class UError* OutError);
void EventResult(class UPrivilegeCheck_X* OutResult);
};

// Class TAGame.__TAsyncResult__PrivilegeCheck_X__NotifyOnResult_0x1
// 0x0018 (0x0060 - 0x0078)
class U__TAsyncResult__PrivilegeCheck_X__NotifyOnResult_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__TAsyncResult__PrivilegeCheck_X__NotifyOnResult_0x1");
        }
    }

    return uClassPointer;
};

void __TAsyncResult__PrivilegeCheck_X__NotifyOnResult_0x1();
};

// Class TAGame.__TAsyncResult__PrivilegeCheck_X__NotifyOnResultComplete_0x1
// 0x0018 (0x0060 - 0x0078)
class U__TAsyncResult__PrivilegeCheck_X__NotifyOnResultComplete_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {

```

```
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TAsyncResult_PrivilegeCheck_X_NotifyOnResultComplete_0x1");
}

return uClassPointer;
};

void __TAsyncResult_PrivilegeCheck_X_NotifyOnResultComplete_0x1(class UError* Err);
};

// Class TAGame._TAsyncResult_PrivilegeCheck_X_SetResultWhen_0x1
// 0x0018 (0x0060 - 0x0078)
class U_TAsyncResult_PrivilegeCheck_X_SetResultWhen_0x1 : public UObject
{
public:
struct FScriptDelegate           GetResultDelegate;          // 0x0060 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TAsyncResult_PrivilegeCheck_X_SetResultWhen_0x1");
}

return uClassPointer;
};

void __TAsyncResult_PrivilegeCheck_X_SetResultWhen_0x1();
};

// Class TAGame._TAsyncResult_UserBugReportUploadResult_TA_Copy_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TAsyncResult_UserBugReportUploadResult_TA_Copy_0x1 : public UObject
{
public:
class UTAsyncResult_UserBugReportUploadResult_TA* Instance;      // 0x0060
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class
TAGame._TAsyncResult__UserBugReportUploadResult_TA__Copy_0x1");
}

return uClassPointer;
};

void __TAsyncResult__UserBugReportUploadResult_TA__Copy_0x1(class
UUserBugReportUploadResult_TA* R, class UError* E);
};

// Class TAGame.TAsyncResult__UserBugReportUploadResult_TA
// 0x0050 (0x00D0 - 0x0120)
class UTAsyncResult__UserBugReportUploadResult_TA : public UAsyncTask
{
public:
class UUserBugReportUploadResult_TA*          Result;           // 0x00D0
(0x0008) [0x0001004000000000]
struct FScriptDelegate           __EventResult__Delegate;    // 0x00D8 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventResultComplete__Delegate; // 0x00F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __ResultDelegate__Delegate;   // 0x0108
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.TAsyncResult__UserBugReportUploadResult_TA");
}

return uClassPointer;
};

static class UTAsyncResult__UserBugReportUploadResult_TA* CreateResultError(class UError*
InError);
static class UTAsyncResult__UserBugReportUploadResult_TA* CreateResult(class
UUserBugReportUploadResult_TA* InResult);
class UTAsyncResult__UserBugReportUploadResult_TA* Copy();
void eventClearCallbacks();
class UTAsyncResult__UserBugReportUploadResult_TA* eventSetResultWhen(class
UAsyncTask* Other, struct FScriptDelegate GetResultDelegate);
class UTAsyncResult__UserBugReportUploadResult_TA* eventSetResult(class
UUserBugReportUploadResult_TA* InResult, class UError* InError);
class UTAsyncResult__UserBugReportUploadResult_TA* NotifyOnResultComplete(struct
FScriptDelegate Callback);
class UTAsyncResult__UserBugReportUploadResult_TA* NotifyOnResult(struct FScriptDelegate
Callback);
class UUserBugReportUploadResult_TA* ResultDelegate();

```

```
void EventResultComplete(class UUserBugReportUploadResult_TA* OutResult, class UError* OutError);
void EventResult(class UUserBugReportUploadResult_TA* OutResult);
};

// Class TAGame.UserBugReportUploadResult_TA
// 0x0010 (0x0060 - 0x0070)
class UUserBugReportUploadResult_TA : public UObject
{
public:
class FString ConfirmationCode; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportUploadResult_TA");
}

return uClassPointer;
};

};

// Class TAGame.__TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResult_0x1
// 0x0018 (0x0060 - 0x0078)
class U__TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResult_0x1 : public UObject
{
public:
struct FScriptDelegate Callback; // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResult_0x1");

}

return uClassPointer;
};

void __TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResult_0x1();
};

// Class
```

```
TAGame.__TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResultComplete_0x1
// 0x0018 (0x0060 - 0x0078)
class U__TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResultComplete_0x1 : public
UObject
{
public:
struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResultComplete_0x1");

}

return uClassPointer;
};

void __TAsyncResult__UserBugReportUploadResult_TA__NotifyOnResultComplete_0x1(class
UError* Err);
};

// Class TAGame.__TAsyncResult__UserBugReportUploadResult_TA__SetResultWhen_0x1
// 0x0018 (0x0060 - 0x0078)
class U__TAsyncResult__UserBugReportUploadResult_TA__SetResultWhen_0x1 : public UObject
{
public:
struct FScriptDelegate           GetResultDelegate;    // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__TAsyncResult__UserBugReportUploadResult_TA__SetResultWhen_0x1");
}

return uClassPointer;
};

void __TAsyncResult__UserBugReportUploadResult_TA__SetResultWhen_0x1();
};

// Class TAGame.__Team_TA__GetPartyFollowerIds_0x1
// 0x0008 (0x0060 - 0x0068)
```

```
class U_Team_TA__GetPartyFollowerIds_0x1 : public UObject
{
public:
class APRI_TA*          PRI;           // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Team_TA__GetPartyFollowerIds_0x1");
}

return uClassPointer;
};

bool __Team_TA__GetPartyFollowerIds_0x1(class APRI_TA* P);
};

// Class TAGame.__Team_TA__OnPlayerMarkedAsIdleBanned_0x1
// 0x0008 (0x0060 - 0x0068)
class U_Team_TA__OnPlayerMarkedAsIdleBanned_0x1 : public UObject
{
public:
class APRI_TA*          IdlePlayer;    // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__Team_TA__OnPlayerMarkedAsIdleBanned_0x1");
}

return uClassPointer;
};

bool __Team_TA__OnPlayerMarkedAsIdleBanned_0x1(class APRI_TA* M);
};

// Class TAGame.__Team_TA__UpdateCustomColors_0x2
// 0x0004 (0x0060 - 0x0064)
class U_Team_TA__UpdateCustomColors_0x2 : public UObject
{
public:
unsigned long             bHasCustomColor : 1;      // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Team_TA__UpdateCustomColors_0x2");
}

return uClassPointer;
};

void __Team_TA__UpdateCustomColors_0x2(class ATeam_TA* Team);

// Class TAGame.TTeamNameComponent_TA
// 0x0060 (0x0070 - 0x00D0)
class UTeamNameComponent_TA : public UComponent
{
public:
class FString SanitizedTeamName; // 0x0070 (0x0010)
[0x0008000000400000] (CPF_NeedCtorLink)
unsigned long bHasCustomColor : 1; // 0x0080 (0x0004)
[0x0008000000000000] [0x00000001]
int32_t TeamIndex; // 0x0084 (0x0004)
[0x0000000000000000]

struct FScriptDelegate _EventNameChanged__Delegate; // 0x0088
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _SanitizedTeamName__ChangeNotify; // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bHasCustomColor__ChangeNotify; // 0x00B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TTeamNameComponent_TA");
}

return uClassPointer;
};

void __TeamNameComponent_TA__SetCustomTeamName_0x1(class FString _, class FString InSanitized);
void __bHasCustomColor__ChangeNotifyFunc();
void __SanitizedTeamName__ChangeNotifyFunc();
class FString GetTeamName();
void HandleClubDetails(class UClubDetails_X* Club);

```

```
void SetCustomTeamName(class ATeam_TA* InTeam, class FString NewName);
void UpdateTeamName();
void eventConstruct();
void EventNameChanged(class UTeamNameComponent_TA* TeamNameComponent, class
FString NewName);
};

// Class TAGame._TeamDemoAudioComponent_TA__HandlePreparingDemoFX_0x1
// 0x0028 (0x0060 - 0x0088)
class U_TeamDemoAudioComponent_TA__HandlePreparingDemoFX_0x1 : public UObject
{
public:
struct FDemolishData           InDemoData;          // 0x0060 (0x0028)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TeamDemoAudioComponent_TA__HandlePreparingDemoFX_0x1");
}

return uClassPointer;
};

bool __TeamDemoAudioComponent_TA__HandlePreparingDemoFX_0x1(class APRI_TA* P);
};

// Class TAGame.TeamDemoAudioComponent_TA
// 0x001C (0x00A4 - 0x00C0)
class UTeamDemoAudioComponent_TA : public UActorComponent_X
{
public:
struct FScriptDelegate           __CanUseTeamDemoSounds__Delegate;      // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TeamDemoAudioComponent_TA");
}

return uClassPointer;
};

void HandlePreparingDemoFX(struct FDemolishData InDemoData, class AExplosion_X*
```

```

OutDemoExplosion);
void eventDetached();
void eventAttached();
bool CanUseTeamDemoSounds(struct FDemolishData InDemoData);
};

// Class TAGame.__TourBracketCache_TA__Request_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TourBracketCache_TA__Request_0x1 : public UObject
{
public:
    uint64_t TournamentID; // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__TourBracketCache_TA__Request_0x1");
        }
    }

    return uClassPointer;
};

void __TourBracketCache_TA__Request_0x1(class UError* E);
};

// Class TAGame.TourBracketCache_TA
// 0x0060 (0x0060 - 0x00C0)
class UTourBracketCache_TA : public UObject
{
public:
    class UOnlineGameTournaments_TA* Tournaments; // 0x0060
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UTourConfig_TA* Config; // 0x0068 (0x0008)
    [0x0001800000000001] (CPF_Edit)
    TArray<struct FBracketResult> Brackets; // 0x0070 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)
    TArray<struct FBracketRequest> Requests; // 0x0080 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __OnBracketRetrieved__Delegate; // 0x0090
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __OnBracketError__Delegate; // 0x00A8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
    }
}

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.TourBracketCache_TA");
}

return uClassPointer;
};

bool __TourBracketCache_TA__RequestCached_0x1(struct FTourMatch M);
void __TourBracketCache_TA__Dispose_0x1(struct FBracketRequest R);
void eventDispose();
int32_t GetBracketIndex(uint64_t TournamentID);
void HandleDownload(struct FTourBracket Bracket);
void ClearOnBracketChanged(struct FScriptDelegate Callback);
void NotifyOnBracketChanged(uint64_t TournamentID, struct FScriptDelegate Callback);
void HandleGetBracketError(uint64_t TournamentID, class UError* Error);
void NotifyOnBracketError(uint64_t TournamentID, struct FScriptDelegate Callback);
class UAsyncTask* Request(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct FScriptDelegate Callback);
class UAsyncTask* RequestCached(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct FScriptDelegate Callback);
void OnBracketError(uint64_t TournamentID, class UError* Error);
void OnBracketRetrieved(struct FTourBracket& Bracket);
};

// Class TAGame.TourConfig_TA
// 0x00A0 (0x0078 - 0x0118)
class UTourConfig_TA : public UOnlineConfig_X
{
public:
unsigned long           bEnableUI : 1;           // 0x0078 (0x0004)
[0x0001000040000001] [0x00000001] (CPF_Edit | CPF_EditInlineNotify)
unsigned long           bNextMatchPushNotificationsEnabled : 1;    // 0x0078
(0x0004) [0x0001000000000001] [0x00000002] (CPF_Edit)
unsigned long           bEnableRewards : 1;        // 0x0078 (0x0004)
[0x0001000040000001] [0x00000004] (CPF_Edit | CPF_EditInlineNotify)
unsigned long           bHideBetaUI : 1;         // 0x0078 (0x0004)
[0x0001000040000001] [0x00000008] (CPF_Edit | CPF_EditInlineNotify)
unsigned long           bAllowTeamMemberCheckIns : 1;   // 0x0078
(0x0004) [0x0001000000000000] [0x00000010]
int32_t                 CheckInWindowMinutes;      // 0x007C (0x0004)
[0x0001000000000001] (CPF_Edit)
int32_t                 CreateFutureMinutes;       // 0x0080 (0x0004)
[0x0001000000000001] (CPF_Edit)
int32_t                 ActivityWindowMinutes;     // 0x0084 (0x0004)
[0x0001000000000001] (CPF_Edit)
float                   NextMatchPollIntervalSeconds; // 0x0088 (0x0004)
[0x0001000000000001] (CPF_Edit)
float                   NextMatchPollIntervalSecondsPushEnabled; // 0x008C
(0x0004) [0x0001000000000001] (CPF_Edit)
float                   FirstMatchRequestOffsetMin;    // 0x0090 (0x0004)
[0x0001000000000001] (CPF_Edit)
float                   FirstMatchRequestOffsetMax;    // 0x0094 (0x0004)
[0x0001000000000001] (CPF_Edit)
float                   PausedRpcDelaySeconds;       // 0x0098 (0x0004)

```

```

[0x0001000000000001] (CPF_Edit)
int32_t CreateFutureMaxDays; // 0x009C (0x0004)
[0x0001000040000001] (CPF_Edit | CPF_EditInlineNotify)
float BracketCacheSeconds; // 0x00A0 (0x0004)
[0x0001000000000001] (CPF_Edit)
TArray<int32_t> AllowedTeamSizes; // 0x00A8 (0x0010)
[0x0001000004000001] (CPF_Edit | CPF_NeedCtorLink)
class FString AllowedTeamSizesCSV; // 0x00B8 (0x0010)
[0x0001000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t MaxBracketSize; // 0x00C8 (0x0004)
[0x0001000040000001] (CPF_Edit | CPF>EditInlineNotify)
TArray<struct FName> TeamSizeToPlaylistMap; // 0x00D0
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)
TArray<struct FName> GameModeToPlaylistMap; // 0x00E0
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)
uint8_t MaxJoinServerAttempts; // 0x00F0 (0x0001)
[0x000100000000000]
uint8_t JoinServerAttemptsMessagingLimit; // 0x00F1 (0x0001)
[0x000100000000000]
float MatchmakingDelay; // 0x00F4 (0x0004)
[0x0001000000000001] (CPF_Edit)
int32_t MinCheckInDelaySeconds; // 0x00F8 (0x0004)
[0x000100000000000]
int32_t MaxCheckInDelaySeconds; // 0x00FC (0x0004)
[0x000100000000000]
int32_t CheckInDelaySeconds; // 0x0100 (0x0004)
[0x0001004000002000] (CPF_Transient)
TArray<struct FName> PlaylistsWithDynamicMapsets; // 0x0108
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourConfig_TA");
}

return uClassPointer;
};

float GetPausedPollDelay();
bool PlaylistHasDynamicMapset(struct FName PlaylistName);
struct FName GetMapSetNameForGameMode(int32_t GameMode, int32_t TeamSize);
struct FName GetPlaylistName(int32_t GameMode, int32_t TeamSize);
void Apply();
};

// Class TAGame.__TourBracketCache_TA__RequestCached_0x2
// 0x0020 (0x0060 - 0x0080)
class U_TourBracketCache_TA__RequestCached_0x2 : public UObject
{

```

```
public:
int32_t           BracketIndex;          // 0x0060 (0x0004)
[0x0001000000000000]
struct FScriptDelegate      Callback;        // 0x0068 (0x0018)
[0x0001000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TourBracketCache_TA__RequestCached_0x2");
}

return uClassPointer;
};

void __TourBracketCache_TA__RequestCached_0x2();

// Class TAGame._TourBracketSimulator_TA__RoundStep_0x1
// 0x0004 (0x0060 - 0x0064)
class U_TourBracketSimulator_TA__RoundStep_0x1 : public UObject
{
public:
int32_t           BracketRound;          // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TourBracketSimulator_TA__RoundStep_0x1");
}

return uClassPointer;
};

bool __TourBracketSimulator_TA__RoundStep_0x1(struct FTourMatch M);
};

// Class TAGame.TourUtils_TA
// 0x0000 (0x0060 - 0x0060)
class UTourUtils_TA : public UObject
{
public:
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourUtils_TA");
}

return uClassPointer;
};

static bool IsFirstRound(int32_t MatchID, int32_t NumTeams);
static int32_t GetSiblingMatch(int32_t MatchID);
static int32_t CalcTeamWins(uint64_t TeamID, struct FTourMatch& MatchData);
static int32_t NextRound(int32_t Round);
static struct FIntPair CalcPrevMatchIDs(int32_t Round, int32_t Order);
static struct FWORDPair MakeWORDPair(uint64_t A, uint64_t B);
static struct FIntPair MakeIntPair(int32_t A, int32_t B);
static int32_t CalcNextMatchID(int32_t Round, int32_t Order);
static int32_t CalcNextMatchIDByMatchID(int32_t MatchID);
static bool IsRoundOrderValid(int32_t Round, int32_t Order);
static int32_t GetBracketMatchCount(int32_t NumTeams);
static int32_t GetTotalRoundsForTeams(int32_t NumTeams);
static int32_t GetMaxRoundForTeams(int32_t NumTeams);
static int32_t GetNumByes(int32_t NumTeams);
static int32_t GetTeamCountAtRound(int32_t Round);
static int32_t GetMatchCountForRound(int32_t Round);
static int32_t Max0(int32_t Value);
static int32_t CalcMatchID(int32_t Round, int32_t Order);
static int32_t CalcOrder(int32_t MatchID);
static int32_t CalcRound(int32_t MatchID);
};

// Class TAGame.TourBracketSimulator_TA
// 0x0000 (0x0060 - 0x0060)
class UTourBracketSimulator_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourBracketSimulator_TA");
}

return uClassPointer;
};
```

```

static int32_t __TourBracketSimulator_TA__GetEarliestUnfinishedRound_0x2(struct FTourMatch M);
static bool __TourBracketSimulator_TA__GetEarliestUnfinishedRound_0x1(struct FTourMatch M);
static int32_t __TourBracketSimulator_TA__RoundStep_0x2(struct FTourMatch M);
static void GameSet(int32_t MatchID, int32_t GameIndex, int32_t Team0Score, int32_t Team1Score, int32_t GameTimeSeconds, unsigned long bOverTime, int32_t WinningTeamIdx, struct FTourBracket& Bracket);
static void GameStep(int32_t MatchID, int32_t GameIndex, int32_t WinningTeamIdx, struct FTourBracket& Bracket);
static void AdvanceWinner(int32_t FromMatchID, int32_t WinningTeamIdx, struct FTourBracket& Bracket);
static int32_t PickWinner(int32_t Team0Chance, int32_t Team1Chance);
static void InternalStepMatchGames(int32_t MatchID, int32_t Team0Games, int32_t Team1Games, struct FTourBracket& Bracket);
static void MatchStep(int32_t MatchID, int32_t Team0Games, int32_t Team1Games, struct FTourBracket& Bracket);
static void GetRandScore(int32_t MaxScore, int32_t& Score0, int32_t& Score1);
static void RoundStep(int32_t SeriesLength, struct FTourBracket& Bracket);
static int32_t GetEarliestUnfinishedRound(TArray<struct FTourMatch>& Matches);
static int32_t MaxWorkaround(int32_t A, int32_t B);
};

// Class TAGame.__TourCheckIn_TA__CheckIn_0x1
// 0x0078 (0x0060 - 0x00D8)
class U_TourCheckIn_TA_CheckIn_0x1 : public UObject
{
public:
TArray<struct FUniqueNetId> LocalTeamMemberIDs; // 0x0060
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class UOnlineGameParty_TA* Party; // 0x0070 (0x0008)
[0x0001000000000000]
class UTourSettings_TA* Settings; // 0x0078 (0x0008)
[0x0001000000000000]
struct FUniqueNetId PlayerID; // 0x0080 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> AllTeamMemberIDs; // 0x00C8
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__TourCheckIn_TA__CheckIn_0x1");
}

return uClassPointer;
};

void __TourCheckIn_TA__CheckIn_0x3(class UError* InError);
bool __TourCheckIn_TA__CheckIn_0x1(struct FUniqueNetId P);

```

```

};

// Class TAGame.TournamentMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UTournamentMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TournamentMetrics_TA");
}

return uClassPointer;
};

void CheckIn(uint64_t TournamentID, struct FUniqueNetId PlayerID, TArray<struct FUniqueNetId> PartyMembers, unsigned long bSuccessfullyCheckedIn);
static void RecordCheckIn(uint64_t TournamentID, struct FUniqueNetId PlayerID, TArray<struct FUniqueNetId> PartyMembers, unsigned long bSuccessfullyCheckedIn);
};

// Class TAGame.TourCheckIn_TA
// 0x0010 (0x0060 - 0x0070)
class UTourCheckIn_TA : public UObject
{
public:
class UOnlineGameTournaments_TA* Tournaments; // 0x0060
(0x0008) [0x0001004000002000] (CPF_Transient)
class UAsyncTask* CheckInTask; // 0x0068 (0x0008)
[0x0001004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourCheckIn_TA");
}

return uClassPointer;
};

void __TourCheckIn_TA__CheckIn_0x2(class UError* _);
bool IsCheckingIn();
class UErrorType* GetCheckInError(struct FUniqueNetId PlayerID, class UTourSettings_TA* InSettings, unsigned long bHasTeam);

```

```

void HandleCheckIn(struct FUniqueNetId PlayerID, class UTourSettings_TA* TourSettings, struct
FTourTeam Team);
class UAsyncTask* CheckIn(struct FUniqueNetId PlayerID, class UTourSettings_TA* Settings,
struct FCustomMatchTeamSettings TeamSettings, uint64_t TeamID);
void Init(class UOnlineGameTournaments_TA* InTournaments);
};

// Class TAGame.__TourCheckInError_TA__AlertSignupError_0x1
// 0x0010 (0x0060 - 0x0070)
class U_TourCheckInError_TA__AlertSignupError_0x1 : public UObject
{
public:
class UGFxShell_TA* Shell; // 0x0060 (0x0008)
[0x0001000000000000]
uint64_t TournamentID; // 0x0068 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__TourCheckInError_TA__AlertSignupError_0x1");
}

return uClassPointer;
};

void __TourCheckInError_TA__AlertSignupError_0x2(class UGFxModal_X* _);
void __TourCheckInError_TA__AlertSignupError_0x1(class UGFxModal_X* _);

// Class TAGame.__TourGameUpdateDispatcher_TA__HandleUpdateGameSuccess_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TourGameUpdateDispatcher_TA__HandleUpdateGameSuccess_0x1 : public UObject
{
public:
class UOnlineGameTourServer_TA* TourServer; // 0x0060
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__TourGameUpdateDispatcher_TA__HandleUpdateGameSuccess_0x1");
}
}

```

```

return uClassPointer;
};

void __TourGameUpdateDispatcher_TA__HandleUpdateGameSuccess_0x1(struct
FTeamUpdateInfo T);
};

// Class TAGame.RPC_TourUpdateGame_TA
// 0x00A8 (0x00E8 - 0x0190)
class URPC_TourUpdateGame_TA : public URPC_X
{
public:
class FString MatchGuid; // 0x00E8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString ServerId; // 0x00F8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
uint64_t TournamentID; // 0x0108 (0x0008)
[0x0001000000000000]
int32_t MatchID; // 0x0110 (0x0004)
[0x0001000000000000]
int32_t Game; // 0x0114 (0x0004)
[0x0001000000000000]
struct FTourMatchGame GameData; // 0x0118 (0x0018)
[0x0001000000000000]
uint64_t Winner; // 0x0130 (0x0008)
[0x0001000000000000]
TArray<struct FUniqueNetId> PlayersNoShow; // 0x0138 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> PlayersLeftEarly; // 0x0148 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> PlayersIdle; // 0x0158 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
unsigned long bHasNetworkIssues : 1; // 0x0168 (0x0004)
[0x0001000000000000] [0x00000001]
TArray<struct FTourPlayerStats> PlayersStats; // 0x0170 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FTeamUpdateInfo> Teams; // 0x0180 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourUpdateGame_TA");
}

return uClassPointer;
};

};

```

```

// Class TAGame.TourGameUpdateDispatcher_TA
// 0x00D0 (0x0060 - 0x0130)
class UTourGameUpdateDispatcher_TA : public UObject
{
public:
class UMatchSeries_TA*           MatchSeries;          // 0x0060 (0x0008)
[0x0001800000000001] (CPF_Edit)
class UOnlineGameReservations_X*   Reservations;        // 0x0068
(0x0008) [0x0001800000000001] (CPF_Edit)
class FString                   MatchGuid;           // 0x0070 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                   GameServerID;        // 0x0080 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
uint64_t                         TournamentID;       // 0x0090 (0x0008)
[0x0001004000002000] (CPF_Transient)
int32_t                           MatchID;             // 0x0098 (0x0004)
[0x0001004000002000] (CPF_Transient)
int32_t                           GameNum;            // 0x009C (0x0004)
[0x0001004000002000] (CPF_Transient)
TArray<uint64_t>                 TeamIDs;            // 0x00A0 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FTourMatchGame            GameData;            // 0x00B0 (0x0018)
[0x0009004000002000] (CPF_Transient)
TArray<struct FUniqueNetId>       ReservedPlayerIDs; // 0x00C8
(0x0010) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
uint64_t                           SeriesWinningTeamID; // 0x00D8 (0x0008)
[0x0001004000002000] (CPF_Transient)
TArray<uint64_t>                 PsyNetTeamIdOrder; // 0x00E0 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTourGameUpdate_TA*         GameUpdate;          // 0x00F0
(0x0008) [0x0001004000000000]
int32_t                           InactiveGameTimeValue; // 0x00F8 (0x0004)
[0x0001000000000002] (CPF_Const)
struct FScriptDelegate            __EventUpdateError__Delegate; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate            __GameData__ChangeNotify; // 0x0118
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourGameUpdateDispatcher_TA");
}

return uClassPointer;
};

void __TourGameUpdateDispatcher_TA__Init_0x4(class ATeam_TA* T);
struct FUniqueNetId __TourGameUpdateDispatcher_TA__Init_0x2(struct FTourPlayer P);
TArray<struct FTourPlayer> __TourGameUpdateDispatcher_TA__Init_0x1(struct FTourTeam T);

```

```

void __TourGameUpdateDispatcher_TA__HandleGameEnded_0x1(class ATeam_TA* Team);
void __TourGameUpdateDispatcher_TA__OnUpdated_0x1();
struct FUniqueNetId __TourGameUpdateDispatcher_TA__HandleMatchStarted_0x1(class AController* P);
void __TourGameUpdateDispatcher_TA__SendUpdateGameRPC_0x1(class URPC_X* RPC);
void __GameData__ChangeNotifyFunc();
void HandleUpdateGameSuccess(class URPC_TourUpdateGame_TA* RPC);
void HandleSeriesGameForfeit(class UMatchSeries_TA* InMatchSeries, int32_t WinningTeamIndex);
void SendUpdateGameRPC(int32_t InGameNum, struct FTourMatchGame InGameData, uint64_t SeriesWinnerID);
uint64_t GetTeamID(class ATeam_TA* Team);
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* GameEvent);
int32_t GetTeamScoreIndex(class ATeam_TA* Team);
void HandleTeamScoreUpdated(class ATeam_TA* Team);
void UpdateGameTimeSeconds(int32_t Value);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* GameEvent);
void CheckForNetworkIssues();
void HandleMatchStarted(class AGameEvent_Soccar_TA* GameEvent);
void OnUpdated();
void HandleStartNewRound(class AGameEvent_Soccar_TA* GameEvent);
void HandleGameEnded(class AGameEvent_Soccar_TA* GameEvent);
void Init(class AGameEvent_Soccar_TA* GameEvent, struct FTourServerSettings& Settings, TArray<uint64_t>& InPsyNetTeamIdOrder);
void EventUpdateError(class UError* Error);
};

// Class TAGame.__TourGameUpdateDispatcher_TA__Init_0x3
// 0x00E8 (0x0060 - 0x0148)
class U__TourGameUpdateDispatcher_TA__Init_0x3 : public UObject
{
public:
    struct FTourServerSettings           Settings;           // 0x0060 (0x00E8)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__TourGameUpdateDispatcher_TA__Init_0x3");
        }

        return uClassPointer;
    }

    uint64_t __TourGameUpdateDispatcher_TA__Init_0x3(class ATeam_TA* T);
};

// Class TAGame.__TourList_TA__AddUnique_0x1
// 0x0008 (0x0060 - 0x0068)

```

```
class U_TourList_TA__AddUnique_0x1 : public UObject
{
public:
class UTourSettings_TA*           Tournament;          // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__TourList_TA__AddUnique_0x1");
}

return uClassPointer;
};

bool __TourList_TA__AddUnique_0x1(class UTourSettings_TA* T);
};

// Class TAGame.__TourList_TA__GetTournamentResult_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TourList_TA__GetTournamentResult_0x1 : public UObject
{
public:
uint64_t                          TournamentID;        // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__TourList_TA__GetTournamentResult_0x1");
}

return uClassPointer;
};

bool __TourList_TA__GetTournamentResult_0x1(struct FTourResult R);
};

// Class TAGame.__TourPartyCheckIn_TA__HandleLeaderCheckIn_0x2
// 0x0010 (0x0060 - 0x0070)
class U_TourPartyCheckIn_TA__HandleLeaderCheckIn_0x2 : public UObject
{
public:
class UPartyMessage_TourCheckIn_TA*    Message;          // 0x0060
(0x0008) [0x0001000000000000]
class UTourPartyCheckInTransaction_TA*   Transaction;       // 0x0068
```

(0x0008) [0x0001000000000000]

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._TourPartyCheckIn_TA_HandleLeaderCheckIn_0x2");  
}  
  
return uClassPointer;  
};  
  
void __TourPartyCheckIn_TA_HandleLeaderCheckIn_0x2(class UTourSettings_TA* _);  
};  
  
// Class TAGame.TourPartyCheckInTransaction_TA  
// 0x0040 (0x0060 - 0x00A0)  
class UTourPartyCheckInTransaction_TA : public UObject  
{  
public:  
class UOnlineGameTournaments_TA* Tournaments; // 0x0060  
(0x0008) [0x0001800000000001] (CPF_Edit)  
class UOnlineGameParty_X* Party; // 0x0068 (0x0008)  
[0x0001800000000001] (CPF_Edit)  
class UError* CheckInError; // 0x0070 (0x0008)  
[0x0001004000000000]  
class UTourSettings_TA* Settings; // 0x0078 (0x0008)  
[0x0001004000000000]  
unsigned long bDisposed : 1; // 0x0080 (0x0004)  
[0x0001004000000000] [0x00000001]  
struct FScriptDelegate __EventPartyCheckInError__Delegate; // 0x0088  
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.TourPartyCheckInTransaction_TA");  
}  
  
return uClassPointer;  
};  
  
void NotifyPartyOfError(class UError* Error);  
void SetError(class UError* Error);  
void HandlePartySizeChanged(class UOnlineGameParty_X* InParty, int32_t NewSize, int32_t OldSize);
```

```

void eventDispose();
void Init(class UTourSettings_TA* InSettings);
void EventPartyCheckInError(class UTourSettings_TA* TourSettings, class UError* Error);
};

// Class TAGame.PartyMessage_TourCheckIn_TA
// 0x0070 (0x00A8 - 0x0118)
class UPartyMessage_TourCheckIn_TA : public UPartyMessage_X
{
public:
class UTourSettings_TA*           TourSettings;          // 0x00A8 (0x0008)
[0x0001000000000000]
struct FTourTeam                  Team;                // 0x00B0 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FTourPrivateCredentials     Credentials;        // 0x00F8 (0x0020)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_TourCheckIn_TA");
}

return uClassPointer;
};

class UPartyMessage_TourCheckIn_TA* SetCredentials(struct FTourPrivateCredentials
InCredentials);
class UPartyMessage_TourCheckIn_TA* SetTeam(struct FTourTeam InTeam);
class UPartyMessage_TourCheckIn_TA* SetTourSettings(class UTourSettings_TA* InSettings);
};

// Class TAGame.TourPartyCheckIn_TA
// 0x0028 (0x0060 - 0x0088)
class UTourPartyCheckIn_TA : public UObject
{
public:
class UTourConfig_TA*            TourConfig;          // 0x0060 (0x0008)
[0x0001800000000000]
class FString                     MemberCheckInError; // 0x0068 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UOnlineGameTournaments_TA*   Tournaments;        // 0x0078
(0x0008) [0x0001004000002000] (CPF_Transient)
class UOnlineGameParty_X*          Party;              // 0x0080 (0x0008)
[0x0001004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourPartyCheckIn_TA");
}

return uClassPointer;
};

void __TourPartyCheckIn_TA__HandleLeaderCheckIn_0x1(class UTourSettings_TA* TourSettings,
class UError* Error);
void HandlePreCheckInMessage(class UOnlineMessageComponent_X* Component, class
UPartyMessage_TourPreCheckIn_TA* Message);
void HandleMemberCheckInError(struct FUniqueNetId PartyMemberID, uint64_t TournamentID,
class UError* Error);
void HandleMemberCheckInErrorMessage(class UOnlineMessageComponent_X* Component,
class UPartyMessage_TourCheckInError_TA* Message);
void HandleError(class UError* Error);
void TournamentCheckIn(class UPartyMessage_TourCheckIn_TA* Message, class
UTourPartyCheckInTransaction_TA* Transaction);
void HandleLeaderCheckIn(class UOnlineMessageComponent_X* Component, class
UPartyMessage_TourCheckIn_TA* Message);
void BroadcastTournamentCheckIn();
void Init(class UOnlineGameTournaments_TA* InTournaments, class UOnlineGameParty_X*
InParty);
};

// Class TAGame.TourRegistration_TA
// 0x0030 (0x0060 - 0x0090)
class UTourRegistration_TA : public UObject
{
public:
class UOnlineGameTournaments_TA* Tournaments; // 0x0060
(0x0008) [0x001004000002000] (CPF_Transient)
class UTourSubscriptions_TA* Subscriptions; // 0x0068 (0x0008)
[0x001004000002000] (CPF_Transient)
class UAsyncTask* RegistrationTask; // 0x0070 (0x0008)
[0x001004000002000] (CPF_Transient)
struct FScriptDelegate __OnRegistered__Delegate; // 0x0078
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourRegistration_TA");
}

return uClassPointer;
};

```

```

void __TourRegistration_TA__Register_0x2(class UError* _);
class UErrorType* CheckRegistrationError(class UTourSettings_TA* Settings);
class UAsyncTask* Register(struct FUniqueNetId PlayerID, class UTourSettings_TA* Settings,
struct FTourPrivateCredentials Credentials, struct FScriptDelegate Callback);
void Init(class UOnlineGameTournaments_TA* InTournaments);
void OnRegistered(class UTourSettings_TA* TourSettings);
};

// Class TAGame.__TourRegistration_TA__Register_0x1
// 0x0018 (0x0060 - 0x0078)
class U_TourRegistration_TA_Register_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__TourRegistration_TA__Register_0x1");
        }

        return uClassPointer;
    }

void __TourRegistration_TA__Register_0x1(class UTourSettings_TA* Tournament);
};

// Class TAGame.TourService_TA
// 0x00A8 (0x0060 - 0x0108)
class UTourService_TA : public UObject
{
public:
    struct FScriptDelegate           __OnTournamentResult__Delegate;      // 0x0060
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __OnCheckIn__Delegate;           // 0x0078 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __OnGetBracket__Delegate;         // 0x0090
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __OnGetMatch__Delegate;          // 0x00A8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __OnGetMatchDetails__Delegate;   // 0x00C0
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __OnGetRewards__Delegate;        // 0x00D8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __OnStartMatchmaking__Delegate; // 0x00F0
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
}

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourService_TA");
}

return uClassPointer;
};

class UAsyncTask* StartMatchmaking(struct FUniqueNetId PlayerID, struct FScriptDelegate Callback);
void OnStartMatchmaking();
class UAsyncTask* GetRewards(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct FScriptDelegate Callback);
void OnGetRewards(uint64_t TournamentID, TArray<struct FTourReward> Rewards);
class UAsyncTask* GetMatchDetails(uint64_t TournamentID, int32_t MatchID, struct FScriptDelegate Callback);
void OnGetMatchDetails(TArray<struct FTourMatchTeamDetails>& TeamStats);
class UAsyncTask* GetMatch(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct FScriptDelegate Callback);
void OnGetMatch(struct FTourMatch Match);
class UAsyncTask* GetBracket(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct FScriptDelegate Callback);
void OnGetBracket(struct FTourBracket Bracket);
class UAsyncTask* CheckIn(struct FUniqueNetId PlayerID, struct FUniqueLobbyId PartyID, class UTourSettings_TA* TourSettings, struct FCustomMatchTeamSettings Settings, TArray<struct FUniqueNetId> LocalTeamMemberIDs, TArray<struct FUniqueNetId> AllTeamMemberIDs, uint64_t TeamID, struct FScriptDelegate Callback);
void OnCheckIn(struct FUniqueNetId PlayerID, class UTourSettings_TA* TourSettings, struct FTourTeam Team);
class UAsyncTask* RegisterTournament(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct FTourPrivateCredentials Credentials, struct FScriptDelegate Callback);
class UAsyncTask* CreateTournament(struct FUniqueNetId PlayerID, struct FTourCreateSettings Settings, struct FScriptDelegate Callback);
void OnTournamentResult(class UTourSettings_TA* NewTournament);
};

// Class TAGame.__TourService_PsyNet_TA__CheckIn_0x1
// 0x0050 (0x0060 - 0x00B0)
class U_TourService_PsyNet_TA_CheckIn_0x1 : public UObject
{
public:
    struct FScriptDelegate           Callback;          // 0x0060 (0x0018)
    [0x0001000000400000] (CPF_NeedCtorLink)        class UTourSettings_TA*   TourSettings;      // 0x0078 (0x0008)
    [0x0001000000000000]            struct FCustomMatchTeamSettings Settings; // 0x0080 (0x0020)
    [0x0001000000400000] (CPF_NeedCtorLink)        TArray<struct FUniqueNetId> AllTeamMemberIDs; // 0x00A0
    (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__TourService_PsyNet_TA__CheckIn_0x1");
}

return uClassPointer;
};

void __TourService_PsyNet_TA__CheckIn_0x1(class URPC_TourCheckIn_TA* RPC);
};

// Class TAGame.RPC_TourCheckIn_TA
// 0x00B8 (0x00E8 - 0x01A0)
class URPC_TourCheckIn_TA : public URPC_X
{
public:
uint64_t TournamentID; // 0x00E8 (0x0008)
[0x0001004000000000]
struct FUniqueNetId PlayerID; // 0x00F0 (0x0048)
[0x0001004000400000] (CPF_NeedCtorLink)
struct FUniqueLobbyId PartyID; // 0x0138 (0x0010)
[0x0001004000000000]
struct FCustomMatchTeamSettings Settings; // 0x0148 (0x0020)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> LocalTeamMembers; // 0x0168
(0x0010) [0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> TeamMembers; // 0x0178 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
uint64_t TeamID; // 0x0188 (0x0008)
[0x0001004000000000]
class FString TeamName; // 0x0190 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourCheckIn_TA");
}

return uClassPointer;
};

class URPC_TourCheckIn_TA* SetTeamID(uint64_t InTeamID);
class URPC_TourCheckIn_TA* SetTeamMembers(TArray<struct FUniqueNetId>& InTeamMembers);
class URPC_TourCheckIn_TA* SetLocalTeamMembers(TArray<struct FUniqueNetId>&

```

```

InLocalTeamMembers);
class URPC_TourCheckIn_TA* SetTeamSettings(struct FCustomMatchTeamSettings&
InSettings);
class URPC_TourCheckIn_TA* SetPartyId(struct FUniqueLobbyId InPartyId);
class URPC_TourCheckIn_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
class URPC_TourCheckIn_TA* SetTournamentID(uint64_t InTournamentID);
};

// Class TAGame.TourService_PsyNet_TA
// 0x0000 (0x0108 - 0x0108)
class UTourService_PsyNet_TA : public UTourService_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourService_PsyNet_TA");
}

return uClassPointer;
};

struct FTourPlayer __TourService_PsyNet_TA__BuildTeam_0x1(struct FUniqueNetId Id);
class UAsyncTask* StartMatchmaking(struct FUniqueNetId PlayerID, struct FScriptDelegate
Callback);
class UAsyncTask* GetMatchDetails(uint64_t TournamentID, int32_t MatchID, struct
FScriptDelegate Callback);
class UAsyncTask* GetMatch(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct
FScriptDelegate Callback);
class UAsyncTask* GetBracket(struct FUniqueNetId PlayerID, uint64_t TournamentID, struct
FScriptDelegate Callback);
struct FTourTeam BuildTeam(uint64_t TeamID, class FString FinalizedTeamName, struct
FCustomMatchTeamSettings Settings, TArray<struct FUniqueNetId> Players);
class UAsyncTask* CheckIn(struct FUniqueNetId PlayerID, struct FUniqueLobbyId PartyID, class
UTourSettings_TA* TourSettings, struct FCustomMatchTeamSettings Settings, TArray<struct
FUniqueNetId> LocalTeamMemberIDs, TArray<struct FUniqueNetId> AllTeamMemberIDs,
uint64_t TeamID, struct FScriptDelegate Callback);
class UAsyncTask* RegisterTournament(struct FUniqueNetId PlayerID, uint64_t TournamentID,
struct FTourPrivateCredentials Credentials, struct FScriptDelegate Callback);
class UAsyncTask* CreateTournament(struct FUniqueNetId PlayerID, struct FTourCreateSettings
Settings, struct FScriptDelegate Callback);
};

// Class TAGame.__TourService_PsyNet_TA__CreateTournament_0x1
// 0x0018 (0x0060 - 0x0078)
class U__TourService_PsyNet_TA__CreateTournament_0x1 : public UObject
{
public:
struct FScriptDelegate           Callback;                                // 0x0060 (0x0018)

```

[0x0001000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.__TourService_PsyNet_TA__CreateTournament_0x1");  
}  
  
return uClassPointer;  
};  
  
void __TourService_PsyNet_TA__CreateTournament_0x1(class URPC_TourCreate_TA* RPC);  
};  
  
// Class TAGame.RPC_TourCreate_TA  
// 0x0088 (0x00E8 - 0x0170)  
class URPC_TourCreate_TA : public URPC_X  
{  
public:  
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)  
[0x0001004000400000] (CPF_NeedCtorLink)  
struct FTourCreateSettings     Settings;          // 0x0130 (0x0038)  
[0x0001004000400000] (CPF_NeedCtorLink)  
class UTourSettings_TA*       Tournament;        // 0x0168 (0x0008)  
[0x0001004000002000] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourCreate_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TourCreate_TA* SetSettings(struct FTourCreateSettings& InSettings);  
class URPC_TourCreate_TA* SetPlayerID(struct FUniqueNetId InPlayerId);  
};  
  
// Class TAGame.__TourService_PsyNet_TA__GetBracket_0x1  
// 0x0018 (0x0060 - 0x0078)  
class U__TourService_PsyNet_TA__GetBracket_0x1 : public UObject  
{  
public:  
struct FScriptDelegate         Callback;          // 0x0060 (0x0018)
```

[0x0001000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._TourService_PsyNet_TA__GetBracket_0x1");  
}  
  
return uClassPointer;  
};  
  
void __TourService_PsyNet_TA__GetBracket_0x1(class URPC_TourGetBracket_TA* RPC);  
};  
  
// Class TAGame.RPC_TourGetBracket_TA  
// 0x00C8 (0x00E8 - 0x01B0)  
class URPC_TourGetBracket_TA : public URPC_X  
{  
public:  
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)  
[0x0001004000400000] (CPF_NeedCtorLink)  
uint64_t                      TournamentID;     // 0x0130 (0x0008)  
[0x0001004000000000]  
struct FTourBracket            Bracket;          // 0x0138 (0x0078)  
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetBracket_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TourGetBracket_TA* SetTournamentID(uint64_t InTournamentID);  
class URPC_TourGetBracket_TA* SetPlayerID(struct FUniqueNetId& InPlayerId);  
};  
  
// Class TAGame._TourService_PsyNet_TA__GetMatch_0x1  
// 0x0018 (0x0060 - 0x0078)  
class U_TourService_PsyNet_TA__GetMatch_0x1 : public UObject  
{  
public:  
struct FScriptDelegate          Callback;         // 0x0060 (0x0018)
```

[0x0001000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.__TourService_PsyNet_TA__GetMatch_0x1");  
}  
  
return uClassPointer;  
};  
  
void __TourService_PsyNet_TA__GetMatch_0x1(class URPC_TourGetMatch_TA* RPC);  
};  
  
// Class TAGame.RPC_TourGetMatch_TA  
// 0x0080 (0x00E8 - 0x0168)  
class URPC_TourGetMatch_TA : public URPC_X  
{  
public:  
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)  
[0x0001004000400000] (CPF_NeedCtorLink)  
uint64_t TournamentID; // 0x0130 (0x0008)  
[0x0001004000000000]  
struct FTourMatch Match; // 0x0138 (0x0030)  
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetMatch_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TourGetMatch_TA* SetTournamentID(uint64_t InTournamentID);  
class URPC_TourGetMatch_TA* SetPlayerID(struct FUniqueNetId InPlayerId);  
};  
  
// Class TAGame.__TourService_PsyNet_TA__GetMatchDetails_0x1  
// 0x0018 (0x0060 - 0x0078)  
class U__TourService_PsyNet_TA__GetMatchDetails_0x1 : public UObject  
{  
public:  
struct FScriptDelegate Callback; // 0x0060 (0x0018)  
[0x0001000000400000] (CPF_NeedCtorLink)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TourService_PsyNet_TA__GetMatchDetails_0x1");
}

return uClassPointer;
};

void __TourService_PsyNet_TA__GetMatchDetails_0x1(class URPC_TourGetMatchDetails_TA*
RPC);
};

// Class TAGame.RPC_TourGetMatchDetails_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_TourGetMatchDetails_TA : public URPC_X
{
public:
uint64_t TournamentID; // 0x00E8 (0x0008)
[0x0001004000000000]
int32_t MatchID; // 0x00F0 (0x0004)
[0x0001004000000000]
TArray<struct FTourMatchTeamDetails> TeamStats; // 0x00F8
(0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetMatchDetails_TA");
}

return uClassPointer;
};

class URPC_TourGetMatchDetails_TA* SetMatchID(int32_t InMatchID);
class URPC_TourGetMatchDetails_TA* SetTournamentID(uint64_t InTournamentID);
};

// Class TAGame._TourService_PsyNet_TA__RegisterTournament_0x1
// 0x0018 (0x0060 - 0x0078)
class U_TourService_PsyNet_TA__RegisterTournament_0x1 : public UObject
{
public:
struct FScriptDelegate Callback; // 0x0060 (0x0018)

```

[0x0001000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame._TourService_PsyNet_TA__RegisterTournament_0x1");  
}  
  
return uClassPointer;  
};  
  
void __TourService_PsyNet_TA__RegisterTournament_0x1(class URPC_TourRegister_TA* RPC);  
};  
  
// Class TAGame.RPC_TourRegister_TA  
// 0x0078 (0x00E8 - 0x0160)  
class URPC_TourRegister_TA : public URPC_X  
{  
public:  
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)  
[0x0001004000400000] (CPF_NeedCtorLink)  
uint64_t                      TournamentID;      // 0x0130 (0x0008)  
[0x0001004000000000]  
struct FTourPrivateCredentials Credentials;       // 0x0138 (0x0020)  
[0x0001004000400000] (CPF_NeedCtorLink)  
class UTourSettings_TA*        Tournament;        // 0x0158 (0x0008)  
[0x0001004000002000] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourRegister_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TourRegister_TA* SetCredentials(struct FTourPrivateCredentials& InCredentials);  
class URPC_TourRegister_TA* SetTournamentID(uint64_t InTournamentID);  
class URPC_TourRegister_TA* SetPlayerID(struct FUniqueNetId InPlayerId);  
};  
  
// Class TAGame._TourService_PsyNet_TA__StartMatchmaking_0x1  
// 0x0018 (0x0060 - 0x0078)  
class U_TourService_PsyNet_TA__StartMatchmaking_0x1 : public UObject
```

```

{
public:
struct FScriptDelegate           Callback;           // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._TourService_PsyNet_TA_StartMatchmaking_0x1");
}

return uClassPointer;
};

void __TourService_PsyNet_TA_StartMatchmaking_0x1(class URPC_X* RPC);
};

// Class TAGame.TourStatus_TA
// 0x0018 (0x0060 - 0x0078)
class UTourStatus_TA : public UObject
{
public:
TArray<struct FTourStatus>          Statuses;          // 0x0060 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UOnlineGameTournaments_TA*    Tournaments;        // 0x0070
(0x0008) [0x0001800000000001] (CPF>Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourStatus_TA");
}

return uClassPointer;
};

void __TourStatus_TA_Init_0x2(class UOnlineGameTourMatchmaking_TA* _, class UError* Error);
void __TourStatus_TA_Init_0x1(uint64_t Id);
void __TourStatus_TA_HandleSubscriptions_0x1(class UTourSettings_TA* T);
void __TourStatus_TA_HandleTourActivated_0x1(class UTourEvent_TA* InTourEvent);
void __TourStatus_TA_HandleTourActivated_0x2(class UTourEventMatch_TA* EventMatch,
class UError* Error);
void HandleError(uint64_t TournamentID, class UError* Error);
void HandleTourActivated(class UOnlineGameTournaments_TA* Tournament, class

```

```

UTourEvent_TA* TourEvent);
void UpdateStatus(uint8_t Status, uint64_t TournamentID);
int32_t GetStatusIndex(uint64_t TournamentID);
void HandleBracketError(uint64_t TournamentID, class UError* Error);
void HandleSubscriptions(class UTourList_TA* TourList);
uint8_t GetStatus(uint64_t TournamentID);
void Init(class UOnlineGameTourMatchmaking_TA* TourMatchmaking);
};

// Class TAGame.TourEvent_TA
// 0x00E8 (0x0060 - 0x0148)
class UTourEvent_TA : public UObject
{
public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x0060
(0x0008) [0x0001800000000001] (CPF_Edit)
class UOnlineGameTourMatchmaking_TA*        TourMatchmaking;     // 0x0068
(0x0008) [0x0001800000000001] (CPF_Edit)
struct FUniqueNetId                      PlayerID;            // 0x0070 (0x0048)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTourSettings_TA*                   ActiveTour;         // 0x00B8 (0x0008)
[0x0001004000002000] (CPF_Transient)
struct FTourTeam                          Team;                // 0x00C0 (0x0048)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTourEventMatch_TA*                 EventMatch;         // 0x0108 (0x0008)
[0x0001004000002000] (CPF_Transient)
class UTourEventMatchComplete_TA*         MatchComplete;      // 0x0110
(0x0008) [0x0001004000002000] (CPF_Transient)
struct FScriptDelegate                   _EventStartTimeElapsed__Delegate; // 0x0118
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                   _EventDeactivated__Delegate; // 0x0130
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourEvent_TA");
}

return uClassPointer;
};

void __TourEvent_TA__SetActiveTour_0x1(class ULocalPlayer* O, class ULocalPlayer* N);
struct FUniqueNetId __TourEvent_TA__IsPartyInPlayersTourTeam_0x1(struct FLobbyMember M);
bool IsPartyInPlayersTourTeam(struct FActiveLobbyInfo& LobbyInfo);
bool IsPlayerInTourTeam(struct FUniqueNetId UserPlayerID);
static bool IsTerminalError(class UError* Error);
void HandleError(class UError* Error);
void HandleMatchmakingError(class UOnlineGameTourMatchmaking_TA* InMatchMaking, class UError* Error);

```

```

void HandleMatchError(class UTourEventMatch_TA* InEventMatch, class UError* Error);
void HandleUnsubscribe(class UTourSubscriptions_TA* Subscriptions, uint64_t TournamentID);
void DeactivateWithError(class UError* Error);
void Deactivate();
void HandleStartTimeElapsed();
void NotifyOnStartTimeElapsed(struct FScriptDelegate Callback);
void SetActiveTour(struct FUniqueNetId InPlayerId, class UTourSettings_TA* InSettings, struct FTourTeam& InTeam);
void EventDeactivated(class UTourEvent_TA* TourEvent, class UError* Error);
void EventStartTimeElapsed(class UTourEvent_TA* TourEvent);
};

// Class TAGame.OnlineGameTourMatchmaking_TA
// 0x0090 (0x0130 - 0x01C0)
class UOnlineGameTourMatchmaking_TA : public UOnlineGameMatchmakingBase_X
{
public:
    class UOnlineGameTournaments_TA*           Tournaments;           // 0x0130
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UTourConfig_TA*                     Config;               // 0x0138 (0x0008)
    [0x0001800000000001] (CPF_Edit)
    class FString                           FoundServerString;   // 0x0140 (0x0010)
    [0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class FString                           StartSearchFailString; // 0x0150 (0x0010)
    [0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
    class UTourEvent_TA*                   TourEvent;            // 0x0160 (0x0008)
    [0x0001004000002000] (CPF_Transient)
    class UError*                          Error;                // 0x0168 (0x0008)
    [0x0001004000002000] (CPF_Transient)
    unsigned long                         bPlayerReadyForNextMatch : 1; // 0x0170 (0x0004)
    [0x0001000000002000] [0x00000001] (CPF_Transient)
    uint8_t                               JoinServerAttemptCounter; // 0x0174 (0x0001)
    [0x0001000000002000] (CPF_Transient)
    struct FScriptDelegate             __EventError__Delegate; // 0x0178 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate             __EventJoinedGame__Delegate; // 0x0190
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate             __EventFailedToJoinGame__Delegate; // 0x01A8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.OnlineGameTourMatchmaking_TA");
        }

        return uClassPointer;
    }

    void StartMatchmaking();
};

```

```

void HandleError(class UError* InError);
void HandleMatchmakingStartSuccess();
void HandleStartSearch(class UAsyncTask* Task);
void OnReceiveGameServer(struct FServerReservationData Reservation);
void OnFailedToJoinGame(class FString FailReason);
void SetError(class UError* InError);
void CreateMatchmakingUI(class UGFxEngine_X* GFxEngine, class UGFxShell_X* Shell);
void OnFindGameWarning(class FString FailReason);
void OnExit();
void Cancel();
void StartSearch();
void HandleTourEventDeactivated(class UTourEvent_TA* InTourEvent, class UError* InError);
void Deactivate();
void HandleGameDataSelected(int32_t PlaylistId, int32_t MutatorIndex);
void StartDelayedSearch();
void HandleNextMatchReady(class UTourEventMatch_TA* TourEventMatch, struct FTourMatch& Match);
void HandleStartTimeElapsed(class UTourEvent_TA* InTourEvent);
void HandleTourActivated(class UOnlineGameTournaments_TA* InTournaments, class UTourEvent_TA* InTourEvent);
void PlayerIsReadyForNextMatch();
void OnNewGame();
void EventFailedToJoinGame(unsigned long bRetrying);
void EventJoinedGame();
void EventError(class UOnlineGameTourMatchmaking_TA* TourMatchmaking, class UError* InError);
};

// Class TAGame.TourEventMatch_TA
// 0x0170 (0x0060 - 0x01D0)
class UTourEventMatch_TA : public UObject
{
public:
    class UOnlineGameTournaments_TA* Tournaments; // 0x0060
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UTourConfig_TA* Config; // 0x0068 (0x0008)
    [0x0001800000000001] (CPF_Edit)
    class UOnlineGameTourMatchmaking_TA* TourMatchmaking; // 0x0070
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UPsyNet_X* PsyNet; // 0x0078 (0x0008)
    [0x0001800000000000]
    struct FUniqueNetId PlayerID; // 0x0080 (0x0048)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    uint64_t TournamentID; // 0x00C8 (0x0008)
    [0x0001004000002000] (CPF_Transient)
    uint64_t TeamID; // 0x00D0 (0x0008)
    [0x0001004000002000] (CPF_Transient)
    struct FTourMatch DependentMatch; // 0x00D8 (0x0030)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    struct FTourMatch NextMatch; // 0x0108 (0x0030)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    int32_t PrevMatchID; // 0x0138 (0x0004)
    [0x0001004000002000] (CPF_Transient)
    class UError* Error; // 0x0140 (0x0008)
};

```

```
[0x0001004000002000] (CPF_Transient)
class UAsyncTask*           GetMatchTask;          // 0x0148 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString               ReceivedByeTitle;     // 0x0150 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString               ReceivedByeBody;      // 0x0160 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate      __EventDependentMatchUpdated__Delegate;    //
0x0170 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventNextMatchReady__Delegate;        // 0x0188
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventError__Delegate;       // 0x01A0 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventCurrentMatchUpdated__Delegate;   // 0x01B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourEventMatch_TA");
}

return uClassPointer;
};

void __TourEventMatch_TA__Init_0x1(class UPsyNetClientService_X* S);
void HandleByePushed(class UPsyNetService_TourReceivedBye_TA* Service);
void UpdateDependentMatch(struct FTourMatch& Match);
void UpdateNextMatch(struct FTourMatch& Match);
bool IsNextMatch(struct FTourMatch& Match);
bool IsMatchOutOfDate(struct FTourMatch& Match);
void OnMatchReceived(struct FTourMatch Match);
void HandleMatchPushed(class UPsyNetService_TourMatchFound_TA* Service);
bool OnTerminalError(class UError* InError);
void HandleGetMatchError(class UError* InError);
void SendRequestDelayed();
void SendRequest(unsigned long bRetrying);
void NotifyOnError(struct FScriptDelegate Callback);
void NotifyOnDependentMatchUpdated(struct FScriptDelegate Callback);
bool IsWaitingForMatchResults();
bool HasNextMatch();
void NotifyOnMatchReady(struct FScriptDelegate Callback);
bool IsFirstRound();
bool IsFinalRound();
void StartNextRound();
void Deactivate();
void Init(struct FUniqueNetId InPlayerId, uint64_t InTournamentID, uint64_t InTeamID);
void EventCurrentMatchUpdated(struct FTourMatch& Match);
void EventError(class UTourEventMatch_TA* TourEventMatch, class UError* InError);
void EventNextMatchReady(class UTourEventMatch_TA* TourEventMatch, struct FTourMatch&
```

```
Match);
void EventDependentMatchUpdated(class UTourEventMatch_TA* TourEventMatch, struct
FTourMatch& Match);
};

// Class TAGame.TourSettingsCache_TA
// 0x0010 (0x0060 - 0x0070)
class UTourSettingsCache_TA : public UObject
{
public:
TArray<class UTourList_TA*>           TourLists;           // 0x0060 (0x0010)
[0x0001004004480008] (CPF_ExportObject | CPF_Component | CPF_NeedCtorLink |
CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourSettingsCache_TA");
}

return uClassPointer;
};

static class UTourSettings_TA* StaticFind(uint64_t TourID);
class UTourSettings_TA* Find(uint64_t TourID);
void Remove(class UTourList_TA* List);
void Add(class UTourList_TA* List);
};

// Class TAGame.__TourStatus_TA__GetStatus_0x1
// 0x0008 (0x0060 - 0x0068)
class U__TourStatus_TA__GetStatus_0x1 : public UObject
{
public:
uint64_t           TournamentID;           // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__TourStatus_TA__GetStatus_0x1");
}

return uClassPointer;
};
```

```
bool __TourStatus_TA__GetStatus_0x1(struct FTourStatus S);
};

// Class TAGame.__TourSubscriptions_TA__GetTournament_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TourSubscriptions_TA__GetTournament_0x1 : public UObject
{
public:
    uint64_t TourID; // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.__TourSubscriptions_TA__GetTournament_0x1");
        }
    }

    return uClassPointer;
};

bool __TourSubscriptions_TA__GetTournament_0x1(class UTourSettings_TA* T);
};

// Class TAGame.__TourUtils_TA__CalcTeamWins_0x1
// 0x0008 (0x0060 - 0x0068)
class U_TourUtils_TA__CalcTeamWins_0x1 : public UObject
{
public:
    uint64_t TeamID; // 0x0060 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.__TourUtils_TA__CalcTeamWins_0x1");
        }
    }

    return uClassPointer;
};

bool __TourUtils_TA__CalcTeamWins_0x1(struct FTourMatchGame G);
};

// Class TAGame.__TriggerCondition_TA__AddTriggers_0x1
// 0x0008 (0x0060 - 0x0068)
```

```
class U_TriggerCondition_TA__AddTriggers_0x1 : public UObject
{
public:
class UTriggerClump_TA*           Clump;                      // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__TriggerCondition_TA__AddTriggers_0x1");
}

return uClassPointer;
};

void __TriggerCondition_TA__AddTriggers_0x1(class UStatusTrigger_X* T);
};

// Class TAGame.__UIConfig_TA__FindXPTier_0x1
// 0x0004 (0x0060 - 0x0064)
class U_UIConfig_TA__FindXPTier_0x1 : public UObject
{
public:
int32_t                           XPAmount;                   // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__UIConfig_TA__FindXPTier_0x1");
}

return uClassPointer;
};

bool __UIConfig_TA__FindXPTier_0x1(int32_t XP);
};

// Class TAGame.UIConfig_TA
// 0x0068 (0x0078 - 0x00E0)
class UUIConfig_TA : public UOnlineConfig_X
{
public:
unsigned long                      bReportServerButton : 1;    // 0x0078 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long                      bTournamentTeamColorSection : 1; // 0x0078
```

```

(0x0004) [0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long          bShowCurrencyRechargeButton : 1;           // 0x0078
(0x0004) [0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long          bCrateRouletteEnabled : 1;           // 0x0078 (0x0004)
[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long          bPsyNetFriendListEnabled : 1;           // 0x0078 (0x0004)
[0x0000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long          bShowStatGraphOptions : 1;           // 0x0078 (0x0004)
[0x0000000000000000] [0x00000020]
unsigned long          bUseCoherentBrowser : 1;           // 0x0078 (0x0004)
[0x0000000000000000] [0x00000040]
unsigned long          bRankedSportsEnabled : 1;           // 0x0078 (0x0004)
[0x0000000040000000] [0x00000080] (CPF_EditInlineNotify)
unsigned long          bShowStarterBundles : 1;           // 0x0078 (0x0004)
[0x0000000040000001] [0x00000100] (CPF_Edit | CPF_EditInlineNotify)
unsigned long          bPurchaseProTier : 1;           // 0x0078 (0x0004)
[0x0000000040000001] [0x00000200] (CPF_Edit | CPF_EditInlineNotify)
unsigned long          bUse1LVLIcon : 1;           // 0x0078 (0x0004)
[0x0000000040000000] [0x00000400] (CPF_EditInlineNotify)
unsigned long          bShowAnthems : 1;           // 0x0078 (0x0004)
[0x0000000040000000] [0x00000800] (CPF_EditInlineNotify)
int32_t                ClubInvitationLimit;           // 0x007C (0x0004)
[0x0000000000000000]
uint8_t                MainMenuBG;                  // 0x0080 (0x0001)
[0x0008000000000000]
TArray<struct FTeamColor>      MainMenuTeamColors;        // 0x0088
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t>          XPDDataRanges;            // 0x0098 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
int32_t                TradeConfirmationLockoutSeconds; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString           SupportURL;                 // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                CurrentActiveSeason;        // 0x00C0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FScriptDelegate   __MainMenuBG__ChangeNotify;    // 0x00C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UIConfig_TA");
}

return uClassPointer;
};

int32_t __UIConfig_TA__Apply_0x1(int32_t L, int32_t R);
void __MainMenuBG__ChangeNotifyFunc();
uint8_t FindXPTier(int32_t XPAmount);

```

```
void Apply();
};

// Class TAGame._UserBugReportTypes_TA
// 0x0000 (0x0060 - 0x0060)
class U_UserBugReportTypes_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._UserBugReportTypes_TA");
}

return uClassPointer;
};

};

// Class TAGame.__UserBugReportConfig_TA__GetCategories_0x1
// 0x0008 (0x0060 - 0x0068)
class U__UserBugReportConfig_TA__GetCategories_0x1 : public UObject
{
public:
struct FName ParentCategory; // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__UserBugReportConfig_TA__GetCategories_0x1");
}

return uClassPointer;
};

bool __UserBugReportConfig_TA__GetCategories_0x1(struct FUserBugReportCategory C);
};

// Class TAGame.UserBugReportConfig_TA
// 0x0040 (0x0078 - 0x00B8)
class UUserBugReportConfig_TA : public UOnlineConfig_X
{
public:
```

```

unsigned long          bEnabled : 1;           // 0x0078 (0x0004)
[0x0001000000004001] [0x00000001] (CPF_Edit | CPF_Config)
TArray<struct FUserBugReportCategory>    Categories;           // 0x0080
(0x0010) [0x000100000404001] (CPF_Edit | CPF_Config | CPF_NeedCtorLink)
TArray<struct FUserBugReportCategory>    OnlineCategories;   // 0x0090
(0x0010) [0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class FString>      MemoryDumplgnoreClassNames; // 0x00A0
(0x0010) [0x000100000404001] (CPF_Edit | CPF_Config | CPF_NeedCtorLink)
int32_t                MaxPropertyStringLength; // 0x00B0 (0x0004)
[0x0001000000004001] (CPF_Edit | CPF_Config)
int32_t                MaxDataSizeKilobytes; // 0x00B4 (0x0004)
[0x0001000000004001] (CPF_Edit | CPF_Config)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportConfig_TA");
}

return uClassPointer;
};

bool __UserBugReportConfig_TA__Apply_0x2(struct FName N);
struct FName __UserBugReportConfig_TA__Apply_0x1(struct FUserBugReportCategory C);
bool __UserBugReportConfig_TA__GetMemoryDumplgnoreClasses_0x1(class UClass* C);
TArray<class UClass*> GetMemoryDumplgnoreClasses();
class UClass* FindClass(class FString ClassName);
bool FilterCategory(struct FUserBugReportCategory Category);
TArray<struct FUserBugReportCategory> GetCategories(struct FName ParentCategory);
struct FUserBugReportCategory GetCategory(struct FName CategoryID);
class FString LocalizeCategory(struct FName CategoryID);
void SetParentType(struct FName CategoryID);
void AddOtherCategory(struct FName ParentId);
void OverrideCategory(struct FUserBugReportCategory Category);
void Apply();
};

// Class TAGame.__UserBugReportConfig_TA__GetCategory_0x1
// 0x0008 (0x0060 - 0x0068)
class U__UserBugReportConfig_TA__GetCategory_0x1 : public UObject
{
public:
struct FName          CategoryID;           // 0x0060 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__UserBugReportConfig_TA__GetCategory_0x1");
}

return uClassPointer;
};

bool __UserBugReportConfig_TA__GetCategory_0x1(struct FUserBugReportCategory C);
};

// Class TAGame.UserBugReportGeneratedData_TA
// 0x0028 (0x0060 - 0x0088)
class UUserBugReportGeneratedData_TA : public UObject
{
public:
class UUserBugReportMetaData_TA*           MetaData;          // 0x0060
(0x0008) [0x0010000000000000]
class UUserBugReportBulkData_TA*           BulkData;         // 0x0068 (0x0008)
[0x0001000000000000]
struct FScriptDelegate           __EventComplete__Delegate; // 0x0070
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportGeneratedData_TA");
}

return uClassPointer;
};

void Init(struct FScriptDelegate CompleteCallback);
void EventComplete(class UUserBugReportGeneratedData_TA* GeneratedData);
};

// Class TAGame.UserBugReportMetaData_TA
// 0x0034 (0x0060 - 0x0094)
class UUserBugReportMetaData_TA : public UObject
{
public:
struct FGuid           AppSessionID;      // 0x0060 (0x0010)
[0x0001000000000000]
struct FGuid           LevelSessionID;    // 0x0070 (0x0010)
[0x0001000000000000]
class FString          MatchGuid;        // 0x0080 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t                MatchSeconds;     // 0x0090 (0x0004)
[0x0001000000000000]

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportMetaData_TA");
}

return uClassPointer;
};

void Init();
};

// Class TAGame.__UserBugReportGeneratedData_TA__Init_0x1
// 0x0018 (0x0060 - 0x0078)
class U__UserBugReportGeneratedData_TA__Init_0x1 : public UObject
{
public:
struct FScriptDelegate           CompleteCallback;          // 0x0060 (0x0018)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__UserBugReportGeneratedData_TA__Init_0x1");
}

return uClassPointer;
};

void __UserBugReportGeneratedData_TA__Init_0x1(class UUserBugReportBulkData_TA* _);
};

// Class TAGame.__UserSettingObserverEventCache_TA__RecordSetting_0x1
// 0x0050 (0x0060 - 0x00B0)
class U__UserSettingObserverEventCache_TA__RecordSetting_0x1 : public UObject
{
public:
struct FUniqueNetId           PlayerID;                // 0x0060 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FName                   Id;                     // 0x00A8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.__UserSettingObserverEventCache_TA__RecordSetting_0x1");
}

return uClassPointer;
};

bool __UserSettingObserverEventCache_TA__RecordSetting_0x1(struct FSettingEvent E);
};

// Class TAGame.UserSettingObserverEventCache_TA
// 0x0030 (0x0070 - 0x00A0)
class UUserSettingObserverEventCache_TA : public UComponent
{
public:
float CacheTimeout; // 0x0070 (0x0004)
[0x0001000000000001] (CPF_Edit)
TArray<struct FSettingEvent> Events; // 0x0078 (0x0010)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __EventCacheTimeout__Delegate; // 0x0088
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserSettingObserverEventCache_TA");
}

return uClassPointer;
};

void __UserSettingObserverEventCache_TA__RecordSetting_0x2();
void Reset();
void RecordSetting(struct FUniqueNetId PlayerID, struct FName Id, class FString Value);
void EventCacheTimeout(class UUserSettingObserverEventCache_TA* Cache);
};

// Class TAGame.__VanityQuery_TA__CompleteRequests_0x1
// 0x0004 (0x0060 - 0x0064)
class U_VanityQuery_TA__CompleteRequests_0x1 : public UObject
{
public:
int32_t l; // 0x0060 (0x0004)
[0x0000000000000000]
};

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._VanityQuery_TA__CompleteRequests_0x1");
}

return uClassPointer;
};

bool __VanityQuery_TA__CompleteRequests_0x1(struct FUniqueNetId PID);
};

// Class TAGame._VanitySetManager_TA__RequestAvatars_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_VanitySetManager_TA__RequestAvatars_0x1 : public UObject
{
public:
struct FUniqueNetId           EmptyID;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._VanitySetManager_TA__RequestAvatars_0x1");
}

return uClassPointer;
};

bool __VanitySetManager_TA__RequestAvatars_0x1(struct FUniqueNetId P);
};

// Class TAGame._VanitySetManager_TA__RequestBanners_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_VanitySetManager_TA__RequestBanners_0x1 : public UObject
{
public:
struct FUniqueNetId           EmptyID;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._VanitySetManager_TA_RequestBanners_0x1");
}

return uClassPointer;
};

bool __VanitySetManager_TA_RequestBanners_0x1(struct FUniqueNetId P);
};

// Class TAGame._VanitySetManager_TA_RequestsBorders_0x1
// 0x0048 (0x0060 - 0x00A8)
class U_VanitySetManager_TA_RequestsBorders_0x1 : public UObject
{
public:
struct FUniqueNetId           EmptyID;           // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._VanitySetManager_TA_RequestsBorders_0x1");
}

return uClassPointer;
};

bool __VanitySetManager_TA_RequestsBorders_0x1(struct FUniqueNetId P);
};

// Class TAGame._VoteActor_TA_GetVotes_0x1
// 0x0008 (0x0060 - 0x0068)
class U_VoteActor_TA_GetVotes_0x1 : public UObject
{
public:
uint8_t           VoteStatus;           // 0x0060 (0x0001)
[0x0000000000000000]
unsigned long      bExcludedIdleBannedPlayers : 1;           // 0x0064 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.__VoteActor_TA__GetVotes_0x1");
}

return uClassPointer;
};

bool __VoteActor_TA__GetVotes_0x1(struct FVoter V);
};

// Class TAGame.__Wallet_TA__GetCurrency_0x1
// 0x0004 (0x0060 - 0x0064)
class U_Wallet_TA__GetCurrency_0x1 : public UObject
{
public:
int32_t InCurrencyID; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Wallet_TA__GetCurrency_0x1");
}

return uClassPointer;
};

bool __Wallet_TA__GetCurrency_0x1(struct FCurrency C);
};

// Class TAGame.__Wallet_TA__GiveCurrencyDrops_0x1
// 0x0018 (0x0060 - 0x0078)
class U_Wallet_TA__GiveCurrencyDrops_0x1 : public UObject
{
public:
unsigned long bIsCrewRefund : 1; // 0x0060 (0x0004)
[0x0000000000000000] [0x00000001]
class FString Message; // 0x0068 (0x0010)
[0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.__Wallet_TA__GiveCurrencyDrops_0x1");
}
}

```

```
return uClassPointer;
};

void __Wallet_TA__GiveCurrencyDrops_0x1(struct FCurrency D);
};

// Class TAGame._CrowdTypes_TA
// 0x0000 (0x0060 - 0x0060)
class U_CrowdTypes_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._CrowdTypes_TA");
}

return uClassPointer;
};

};

// Class TAGame._FreeplayCommandsTypes_TA
// 0x0000 (0x0060 - 0x0060)
class U_FreeplayCommandsTypes_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._FreeplayCommandsTypes_TA");
}

return uClassPointer;
};

};

// Class TAGame._TourNotifications_TA
// 0x0000 (0x0060 - 0x0060)
class U_TourNotifications_TA : public UObject
{
public:
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._TourNotifications_TA");
}

return uClassPointer;
};

// Class TAGame._TrainingTypes_TA
// 0x0000 (0x0060 - 0x0060)
class U_TrainingTypes_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame._TrainingTypes_TA");
}

return uClassPointer;
};

// Class TAGame.Achievement_TA
// 0x0008 (0x0060 - 0x0068)
class UAchievement_TA : public UObject
{
public:
uint8_t Type; // 0x0060 (0x0001)
[0x0000000000000001] (CPF_Edit)
int32_t LocalId; // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Achievement_TA");
}
```

```

}

return uClassPointer;
};

static bool ShouldRegister();

// Class TAGame.Achievement_AG_FarFarAway_TA
// 0x0000 (0x0068 - 0x0068)
class UAchievement_AG_FarFarAway_TA : public UAchievement_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Achievement_AG_FarFarAway_TA");
}

return uClassPointer;
};

static bool UpdateDistanceDriven(class UAchievementManager_TA* Manager, class
UUpdateDistanceDrivenTimer_TA* Timer);
static void Update_FarFarAway(class UUpdateDistanceDrivenTimer_TA* Timer, class
UAchievementManager_TA* Manager, class UAchievementSave_TA* SaveData);
static void HandleCarRemoved(class UAchievementManager_TA* Manager, class ACar_TA* Car);
static void HandleCarAdded(class UAchievementManager_TA* Manager, class ACar_TA* Car,
class UAchievementSave_TA* SaveData);
static void CheckUnlock(class UUpdateDistanceDrivenTimer_TA* Timer, class
UAchievementManager_TA* Manager, class UAchievement_AG_FarFarAway_TA* Achievement);
static bool IsAchieved(class UAchievementManager_TA* Manager);
static bool ShouldRegister();
};

// Class TAGame.AchievementSave_TA
// 0x017C (0x00C8 - 0x0244)
class UAchievementSave_TA : public USaveObject_TA
{
public:
TArray<class FString> BotTeamsPlayed; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class FString> TrainingModesPlayed; // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName> LevelsPlayed; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName> LabsMapsPlayed; // 0x00F8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName> CarsPlayed; // 0x0108 (0x0010)

```

```

[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName> RumbleItemsActivated; // 0x0118 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName> BraveTheElementsLevelsPlayed; // 0x0128
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
float TotalBoostTime; // 0x0138 (0x0004)
[0x0000000000000000]
float TotalTimeOnWall; // 0x013C (0x0004)
float TotalDriveDistanceKM; // 0x0140 (0x0004)
int32_t GoalShotsAny; // 0x0144 (0x0004)
int32_t GoalShots; // 0x0148 (0x0004)
int32_t GoalSaves; // 0x014C (0x0004)
int32_t SavageGoals; // 0x0150 (0x0004)
int32_t GameEventsWon; // 0x0154 (0x0004)
int32_t GamesWonInARow; // 0x0158 (0x0004)
int32_t GameEventsPlayed; // 0x015C (0x0004)
int32_t SpectacularGoals; // 0x0160 (0x0004)
int32_t TotalScoredGoals; // 0x0164 (0x0004)
int32_t TotalShotsBlocked; // 0x0168 (0x0004)
int32_t RandomItemsDropped; // 0x016C (0x0004)
int32_t HighestMVPScore; // 0x0170 (0x0004)
int32_t ChampionshipsWon; // 0x0174 (0x0004)
int32_t RegularSeasonsCompleted; // 0x0178 (0x0004)
int32_t HighestRecordedCertifiedRank; // 0x017C (0x0004)
int32_t BreakoutGoals; // 0x0180 (0x0004)
int32_t BreakoutPlatformsDamaged; // 0x0184 (0x0004)
unsigned long bResetWinningIsWinningSeasonList : 1; // 0x0188
(0x0004) [0x0000000000000000] [0x00000001]
TArray<struct FName> CustomizedSlotNames; // 0x0190
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<float> CheckDistancesDrivenKM; // 0x01A0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> CheckScoredGoals; // 0x01B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> CheckShotsBlocked; // 0x01C0 (0x0010)

```

```

[0x0000000000040000] (CPF_NeedCtorLink)
TArray<int32_t> CheckGameEventsWon; // 0x01D0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> CheckRandomItemsDropped; // 0x01E0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> LeftWingRightWingTeamsPlayed; // 0x01F0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<unsigned long> WinningIsWinningPlayedWithDLC0Cars; // 0x0200
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> CarsCollected; // 0x0210 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t ExhibitionMatchesPlayed; // 0x0220 (0x0004)
[0x0000000000000000]
int32_t RankedMatchesPlayed; // 0x0224 (0x0004)
[0x0000000000000000]
int32_t UnrankedMatchesPlayed; // 0x0228 (0x0004)
[0x0000000000000000]
int32_t PrivateMatchesPlayed; // 0x022C (0x0004)
[0x0000000000000000]
int32_t ReplayEditorOpened; // 0x0230 (0x0004)
[0x0000000000000000]
int32_t CreditsViewed; // 0x0234 (0x0004)
[0x0000000000000000]
int32_t GoalsOrAssists; // 0x0238 (0x0004)
[0x0000000000000000]
int32_t CompletedMatchesWithClubmates; // 0x023C (0x0004)
[0x0000000000000000]
int32_t NewOfflineProductsUnlocked; // 0x0240 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AchievementSave_TA");
}

return uClassPointer;
};

void __AchievementSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
int32_t __AchievementSave_TA__GetVersionDelegates_0x3(class UOnlineProduct_TA* BP);
bool __AchievementSave_TA__GetVersionDelegates_0x2(class UOnlineProduct_TA* Op);
void OnCreate();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
class USaveObject_TA* Reconcile(class USaveObject_TA* OtherSaveObject);
};

// Class TAGame.UpdateDistanceDrivenTimer_TA
// 0x0004 (0x0060 - 0x0064)
class UUpdateDistanceDrivenTimer_TA : public UObject

```

```
{  
public:  
int32_t KMDriven; // 0x0060 (0x0004)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.UpdateDistanceDrivenTimer_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.AchievementSystem_TA  
// 0x0000 (0x0060 - 0x0060)  
class UAchievementSystem_TA : public UObject  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.AchievementSystem_TA");  
}  
  
return uClassPointer;  
};  
  
static void CheckWonMatch(class UMatchWinner_TA* Winner, class UAchievementManager_TA* Manager, class APlayerReplicationInfo* PRI);  
static void UpdateStat(class UAchievementManager_TA* Manager, struct FName StatName, int32_t Count);  
static void UpdateProgress(class UAchievementManager_TA* Manager, class UAchievement_TA* Achievement);  
static void UnlockAchievement(class UAchievementManager_TA* Manager, class UAchievement_TA* Achievement);  
static bool ConditionalUnlock(class UAchievementManager_TA* Manager, class UAchievement_TA* Achievement, unsigned long bAchieved);  
static void InitManager(class UAchievementClassCollection_TA* Collection, class UAchievementManager_TA* Manager);  
static void InitSystem(class UEngine* Engine);  
};
```

```
// Class TAGame.Achievement_AG_Stocked_TA
// 0x0000 (0x0068 - 0x0068)
class UAchievement_AG_Stocked_TA : public UAchievement_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Achievement_AG_Stocked_TA");
}

return uClassPointer;
};

static void CheckUnlock(class UAchievementManager_TA* Manager, class
UAchievement_AG_Stocked_TA* Achievement, class URandomItemsDroppedEvent_TA*
DropEvent);
static int32_t RequiredDrops();
static bool IsAchieved(class UAchievementManager_TA* Manager);
};

// Class TAGame.RandomItemsDroppedEvent_TA
// 0x0000 (0x0060 - 0x0060)
class URandomItemsDroppedEvent_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RandomItemsDroppedEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.Achievement_AG_SuperVictorious_TA
// 0x0000 (0x0068 - 0x0068)
class UAchievement_AG_SuperVictorious_TA : public UAchievement_TA
{
public:

public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Achievement_AG_SuperVictorious_TA");
}

return uClassPointer;
};

static void CheckUnlock(class UAchievementManager_TA* Manager, class
UAchievement_AG_SuperVictorious_TA* Achievement, class
UAchievementsGameEventWon_TA* Event);
static bool IsAchieved(class UAchievementManager_TA* Manager);
};

// Class TAGame.AchievementsGameEventWon_TA
// 0x0000 (0x0060 - 0x0060)
class UAchievementsGameEventWon_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AchievementsGameEventWon_TA");
}

return uClassPointer;
};

};

// Class TAGame.Achievement_AG_TheStreak_TA
// 0x0000 (0x0068 - 0x0068)
class UAchievement_AG_TheStreak_TA : public UAchievement_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Achievement_AG_TheStreak_TA");
}
}
```

```
return uClassPointer;
};

static void GameLost(class UAchievementManager_TA* Manager, class
UAchievement_AG_TheStreak_TA* Achievement, class UAchievementsGameEventLost_TA*
Event);
static void GameWon(class UAchievementManager_TA* Manager, class
UAchievement_AG_TheStreak_TA* Achievement, class UAchievementsGameEventWon_TA*
Event);
static bool IsAchieved(class UAchievementManager_TA* Manager);
};

// Class TAGame.AchievementsGameEventLost_TA
// 0x0000 (0x0060 - 0x0060)
class UAchievementsGameEventLost_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AchievementsGameEventLost_TA");
}

return uClassPointer;
};

};

// Class TAGame.Achievement_AG_Virtuoso_TA
// 0x0000 (0x0068 - 0x0068)
class UAchievement_AG_Virtuoso_TA : public UAchievement_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Achievement_AG_Virtuoso_TA");
}

return uClassPointer;
};

static void CheckUnlock(class UAchievementManager_TA* Manager, class
```

```

UAchievement_AG_Virtuoso_TA* Achievement, class UAchievementUnlockedEvent_TA*
UnlockEvent);
static bool IsAchieved(class UAchievementManager_TA* Manager);
static bool ShouldRegister();
};

// Class TAGame.AchievementUnlockedEvent_TA
// 0x0008 (0x0060 - 0x0068)
class UAchievementUnlockedEvent_TA : public UObject
{
public:
    uint8_t           Type;           // 0x0060 (0x0001)
    [0x0000000000000000]           LocalId;
    int32_t          LocalId;        // 0x0064 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AchievementUnlockedEvent_TA");
        }

        return uClassPointer;
    }
};

// Class TAGame.Achievement_AG_Winner_TA
// 0x0000 (0x0068 - 0x0068)
class UAchievement_AG_Winner_TA : public UAchievement_TA
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.Achievement_AG_Winner_TA");
        }

        return uClassPointer;
    }

    static void CheckUnlock(class UAchievementManager_TA* Manager, class
UAchievement_AG_Winner_TA* Achievement, class UAchievementsGameEventWon_TA* Event);
    static bool IsAchieved(class UAchievementManager_TA* Manager);
};

```

```
// Class TAGame.AchievementClassCollection_TA
// 0x0010 (0x0060 - 0x0070)
class UAchievementClassCollection_TA : public UObject
{
public:
TArray<class UClass*>          Classes;           // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AchievementClassCollection_TA");
}

return uClassPointer;
};

};

// Class TAGame.PlayerVanitySave_TA
// 0x0048 (0x00C8 - 0x0110)
class UPlayerVanitySave_TA : public USaveObject_TA
{
public:
struct FProductInstanceID      InstanceID;        // 0x00C8 (0x0010)
[0x0000000000000000]
int32_t                         ProductID;         // 0x00D8 (0x0004)
[0x0000000000000000]
int32_t                         SelectedColorValue; // 0x00DC (0x0004)
[0x0008000000000000]
struct FScriptDelegate          __EventVanityChanged__Delegate; // 0x00E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __SelectedColorValue__ChangeNotify; // 0x00F8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerVanitySave_TA");
}

return uClassPointer;
};

void __SelectedColorValue__ChangeNotifyFunc();
```

```
 TArray<class UProductAttribute_TA*> GetAttributes();
void SetSelectedColorValue(int32_t InColor);
void SetProductInstance(struct FProductInstanceId InInstanceld, int32_t InProductID);
void GetDuplicateReplacementProduct(class USaveData_TA*& SaveData);
void Validate(class UProfile_TA* ForProfile);
void EventVanityChanged(struct FProductInstanceId InInstanceld);
class UProductSlot_TA* GetPlayerSlot();
int32_t GetDefaultProductID();
};

// Class TAGame.PlayerAvatarBorderSave_TA
// 0x0000 (0x0110 - 0x0110)
class UPlayerAvatarBorderSave_TA : public UPlayerVanitySave_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerAvatarBorderSave_TA");
}

return uClassPointer;
};

class UProductSlot_TA* GetPlayerSlot();
int32_t GetDefaultProductID();
};

// Class TAGame.PlayerBannerSave_TA
// 0x0000 (0x0110 - 0x0110)
class UPlayerBannerSave_TA : public UPlayerVanitySave_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerBannerSave_TA");
}

return uClassPointer;
};

class UProductSlot_TA* GetPlayerSlot();
int32_t GetDefaultProductID();
```

```
};
```

```
// Class TAGame.GameInfo_Replay_TA
// 0x0068 (0x04D8 - 0x0540)
class AGameInfo_Replay_TA : public AGameInfoBase_TA
{
public:
    class UReplay_TA*           Replay;           // 0x04D8 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class FString               ReplayPath;       // 0x04E0 (0x0010)
    [0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    class AGameEvent_TA*        GameEvent;        // 0x04F0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    float                      PrevTime;        // 0x04F8 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float                      ReplayStartTime; // 0x04FC (0x0004)
    [0x0000000000002000] (CPF_Transient)
    class FString               ReplayKismetLevelName; // 0x0500 (0x0010)
    [0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    struct FScriptDelegate      __EventReplayStarted__Delegate; // 0x0510
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate      __EventGameEventSet__Delegate; // 0x0528
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Replay_TA");
        }

        return uClassPointer;
    };

    void HandleReplayTimeSkip(class UReplay_TA* InReplay);
    void TriggerTimeEvents();
    void OnLoadReplayCancel(class UGFxModal_X* Modal);
    void OnLoadReplayRetry(class UGFxModal_X* Modal);
    void HandleReplayImportError(class UReplay_TA* InReplay, class UError* Error);
    void eventTick(float DeltaTime);
    void HandleReplayActorSpawned(class UReplay_TA* InReplay, class AActor* A);
    class ANavigationPoint* FindPlayerStart(class AController* Player, uint8_t InTeam, class FString IncomingName);
    void HandleReplayImported(class UReplayManager_TA* Manager, class UReplay_TA* InReplay,
                           class UError* Error);
    void ImportReplay();
    void eventDestroyed();
    void eventInitGame(class FString Options, class FString& ErrorMessage);
    void EventGameEventSet(class AGameInfo_Replay_TA* Game);
    void EventReplayStarted(class AGameInfo_Replay_TA* Game);
};
```

```

// Class TAGame.GameEvent_Season_TA
// 0x0098 (0x0CA0 - 0x0D38)
class AGameEvent_Season_TA : public AGameEvent_Soccar_TA
{
public:
int32_t           PreMatchTime;           // 0x0CA0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class USeason_TA*          Season;        // 0x0CA8 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FSeasonMatch         Match;         // 0xCB0 (0x001C)
[0x0000000000002000] (CPF_Transient)
class UProfile_TA*         Profile;       // 0xCD0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ATeam_TA*            HumanTeam;     // 0xCD8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ATeam_TA*            BotTeam;       // 0xCE0 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t           HumanMatchTeam;        // 0xCE8 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t           BotMatchTeam;          // 0xCEC (0x0004)
[0x0000000000002000] (CPF_Transient)
float             SeasonIntroTime;        // 0xCF0 (0x0004)
[0x0000000000000000]
unsigned long      bIsChampionShipGame : 1; // 0xCF4 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long      bIsFirstGameOfSeason : 1; // 0xCF4 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long      bLastGameOfRegularSeason : 1; // 0xCF4
(0x0004) [0x0000000000000000] [0x00000004]
class USimilarSeasonLogos_TA*    SimilarLogoGroups; // 0xCF8
(0x0008) [0x0000000000000000]
class FString        TeamOverrides[0x2]; // 0xD00 (0x0020)
[0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate    __EventWonChampionShip__Delegate; // 0xD20
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Season_TA");
}

return uClassPointer;
};

void BeginHighlightsReplay();
void OnMatchWinnerSet();
bool GiveExtraXP();
bool AllowSplitScreenPlayer();

```

```

void CheckTeamOverride(int32_t TeamIdx, struct FSeasonTeam& TeamData);
bool IsLastRegularGameOfSeason();
bool IsFirstGameOfSeason();
bool IsChampionshipGame();
void StartGame();
void SaveSeasonStats();
bool ShouldSwapBotColors(class UProductAsset_Logo_TA* PlayerLogo, class
UProductAsset_Logo_TA* BotLogo);
class UProductAsset_Bot_TA* ChooseBotAsset(class AAIController_TA* Bot);
void InitSeasonTeams();
void OnAllTeamsCreated();
void AssignCustomTeamSettings();
class ATeam_TA* PickTeam(class AController* C);
void ToggleDropTimers(unsigned long bEnableTimers);
void PreloadBots();
void OnInit();
void InitGame(class FString Options);
void UpdateStats();
void EventWonChampionShip(class AGameEvent_Season_TA* GameEvent);
};

// Class TAGame.GameInfo_Basketball_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_Basketball_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Basketball_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeasonSave_TA
// 0x0010 (0x00C8 - 0x00D8)
class USeasonSave_TA : public USaveObject_TA
{
public:
TArray<class USeason_TA*> Seasons; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeasonSave_TA");
}

return uClassPointer;
};

void __SeasonSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __SeasonSave_TA__Reconcile_0x1(class USeason_TA* Season);
class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
void OnLoad();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
class USeason_TA* GetLatestSeason();
};

// Class TAGame.Season_TA
// 0x0078 (0x0060 - 0x00D8)
class USeason_TA : public UObject
{
public:
int32_t NumTeams; // 0x0060 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t TeamSize; // 0x0064 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Difficulty; // 0x0068 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t NumRegularWeeks; // 0x006C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t NumPlayoffsWeeks; // 0x0070 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t NumPlayoffsTeams; // 0x0074 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
TArray<struct FSeasonTeam> Teams; // 0x0078 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FSeasonPlayer> Players; // 0x0088 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FSeasonPlayerStat> PlayerStats; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t Year; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t CurrentWeek; // 0x00AC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
TArray<struct FSeasonMatch> Matches; // 0x00B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventWeekSimulated__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Season_TA");
}

return uClassPointer;
};

void PrintRankings(TArray<struct FSeasonTeamRank>& Rankings);
void SetDifficulty(int32_t NewDifficulty);
void SetBots(class FString BotID0, class FString BotName0, class FString BotID1, class FString BotName1, class FString BotID2, class FString BotName2);
void BuildPlayoffsMatches();
void SelectPlayoffsTeams();
void SimulateWeek(unsigned long bForceHumansToWin);
void SaveMatch(struct FSeasonMatch Match);
bool GetMatch(int32_t Team, int32_t Week, unsigned long bReturnByes, struct FSeasonMatch& Match, int32_t& MatchIdx);
int32_t SortTeamRank(struct FSeasonTeamRank A, struct FSeasonTeamRank B);
void GetTeamRankings(int32_t Week, TArray<struct FSeasonTeamRank>& Rankings);
int32_t GetWins(int32_t Team, int32_t Week);
int32_t GetTotalSeasonWeeks();
bool IsFinished(int32_t Week);
int32_t GetLastWeek();
bool IsChampionship(int32_t Week);
bool IsPlayoffs(int32_t Week);
bool IsPostSeason(int32_t Week);
bool IsLastRegularSeason(int32_t Week);
bool IsRegularSeason(int32_t Week);
bool IsFirstGameOfSeason(int32_t Week);
void BuildBotTeams();
void BuildMatches();
void EventWeekSimulated(class USeason_TA* InSeason);
};

// Class TAGame.GameInfo_Breakout_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_Breakout_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Breakout_TA");
}

return uClassPointer;
};

```

```

};

// Class TAGame.GameEvent_Training_TA
// 0x0018 (0xEE0 - 0xEF8)
class AGameEvent_Training_TA : public AGameEvent_Tutorial_TA
{
public:
struct FScriptDelegate           __EventTrainingCompleted__Delegate;      // 0xEE0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Training_TA");
}

return uClassPointer;
};

bool IsPerfectRound();
void BeginHighlightsReplay();
void EventTrainingCompleted(class AGameEvent_Training_TA* Training);
};

// Class TAGame.CarDistanceTracker_TA
// 0x0018 (0x008C - 0x00A4)
class UCarDistanceTracker_TA : public UTickableStateObject_X
{
public:
struct FVector                  LastCarLocation;                      // 0x0090 (0x000C)
[0x0000000000000000]
float                           TotalDriveDistanceKM;                // 0x009C (0x0004)
float                           CurrentDriveDistanceKM;             // 0x00A0 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarDistanceTracker_TA");
}

return uClassPointer;
};

float ConsumeCurrentDriveDistance();

```

```

float ConvertUnrealToKM(float Distance);
void eventTick(float DeltaTime);
void HandlePawnTypeChanged(class APRI_TA* PRI);
void eventConstruct();
};

// Class TAGame.MapPrefsSave_TA
// 0x0030 (0x00C8 - 0x00F8)
class UMapPrefsSave_TA : public USaveObject_TA
{
public:
TArray<struct FPlaylistMapPrefs> MapPrefs; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FName SelectedFreeplayMap; // 0x00D8 (0x0008)
[0x0000000000000000]
struct FScriptDelegate __EventUsedAllMapVotes__Delegate; // 0x00E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapPrefsSave_TA");
}

return uClassPointer;
};

void ClearRemovedMapPrefs(struct FName PlaylistId, TArray<class UMapData_TA*>
PlaylistMaps);
void RecordMapPrefsMetrics();
void SetMapPrefs(struct FPlaylistMapPrefs Prefs, unsigned long bUsedAllVotes);
struct FPlaylistMapPrefs GetMapPrefs(struct FName Playlist);
void EventUsedAllMapVotes(class UMapPrefsSave_TA* MapPrefsSave);
};

// Class TAGame.ProfileStatsSave_TA
// 0x0038 (0x00C8 - 0x0100)
class UProfileStatsSave_TA : public USaveObject_TA
{
public:
TArray<struct FStatValue> StatValues; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductStat> ProductStats; // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventStatValuesChanged__Delegate; // 0x00E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfileStatsSave_TA");
}

return uClassPointer;
};

void OnStatValuesChanged();
void AddProductStatValue(int32_t ProductID, int32_t delta);
int32_t GetProductStatValue(int32_t ProductID);
int32_t GetStatValue(uint8_t StatType, struct FName Id);
void IncrementStatValue(uint8_t StatType, struct FName Id, int32_t Amount);
void SetStatValue(uint8_t StatType, struct FName Id, int32_t Value);
void EventStatValuesChanged(class UProfileStatsSave_TA* StatsSave);
};

// Class TAGame.GameEvent_Breakout_TA
// 0x0018 (0x0CA0 - 0x0CB8)
class AGameEvent_Breakout_TA : public AGameEvent_Soccar_TA
{
public:
unsigned long          bOnlyResetScoredOnTeam : 1;           // 0x0CA0 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bResetOvertime : 1;                 // 0x0CA0 (0x0004)
[0x0000000000000000] [0x00000002]
uint8_t                LastScoredOnTeam;                  // 0x0CA4 (0x0001)
[0x0000000000000000]
TArray<class ABreakOutActor_Platform_TA*>    Platforms;           // 0x0CA8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Breakout_TA");
}

return uClassPointer;
};

void __GameEvent_Breakout_TA__BeginState_0x1(class ABreakOutActor_Platform_TA*
Platform);
void __GameEvent_Breakout_TA__EndState_0x1(class ABreakOutActor_Platform_TA* Platform);
void HandleDamageStateChanged(class ABreakOutActor_Platform_TA* Platform, struct
FBreakoutDamageState InState);
void HandleGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class
UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIndex);
bool ShouldResetPlatform(class ABreakOutActor_Platform_TA* Platform);

```

```

void eventPostBeginPlay();
};

// Class TAGame.MatchType_AutoTournament_TA
// 0x0008 (0x00D0 - 0x00D8)
class UMatchType_AutoTournament_TA : public UMatchType_Tournament_TA
{
public:
class UAutoTourConfig_TA*           AutoTourConfig;          // 0x00D0 (0x0008)
[0x0001800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_AutoTournament_TA");
}

return uClassPointer;
};

bool CanUpdateStats();
bool ShouldSubmitMatchComplete();
void ForcePartyUp(class AGameEvent_Soccar_TA* InGameEvent);
void OnInitGameEvent();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
};

// Class TAGame.VersionHelper_TA
// 0x0030 (0x0060 - 0x0090)
class UVersionHelper_TA : public UObject
{
public:
struct FScriptDelegate           _VersionDelegate__Delegate;    // 0x0060
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           _VersionChangedDelegate__Delegate; // 0x0078
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VersionHelper_TA");
}

return uClassPointer;
};

```

```
static void Update(class UObject* VersionObject, struct FScriptDelegate
VersionChangedCallback, int32_t VersionMax, TArray<struct FScriptDelegate>&
VersionDelegates, int32_t& Version);
void VersionChangedDelegate();
void VersionDelegate(class UObject* VersionObject);
};

// Class TAGame.ReconcileUtil_TA
// 0x0000 (0x0060 - 0x0060)
class UReconcileUtil_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ReconcileUtil_TA");
}

return uClassPointer;
};

static void ReconcileUniqueArrayName(TArray<struct FName>& Destination, TArray<struct FName>& Source);
static void ReconcileUniqueArrayList(TArray<class FString>& Destination, TArray<class FString>& Source);
static void ReconcileQMax(uint64_t& Left, uint64_t& Right);
static void ReconcileMax(int32_t& Left, int32_t& Right);
static void ReconcileFMax(float& Left, float& Right);
};

// Class TAGame.MatchWinner_TA
// 0x0008 (0x0060 - 0x0068)
class UMatchWinner_TA : public UObject
{
public:
class ATeam_TA*           WinningTeam;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchWinner_TA");
}

return uClassPointer;
};
```

```
};

};

// Class TAGame.ActiveGameState_TA
// 0x0000 (0x0060 - 0x0060)
class UActiveGameState_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ActiveGameState_TA");
}

return uClassPointer;
};

};

// Class TAGame.ActivityFeedManager_TA
// 0x0000 (0x0060 - 0x0060)
class UActivityFeedManager_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ActivityFeedManager_TA");
}

return uClassPointer;
};

void __ActivityFeedManager_TA__Construct_0x1(class UAchievementManager_TA* AM);
TArray<class FString> GetFriendsPlayedAgainst(class APRI_TA* OwnerPRI, class AGameEvent_TA* GameEvent);
void CheckFriendsPlayedAgainst(class APRI_TA* OwnerPRI, class AGameEvent_Soccar_TA* GameEvent);
void HandleMatchWinnerSet(class AGameEvent_Soccar_TA* GameEvent);
void HandleChampionshipAmountChanged(class UAchievementManager_TA* Manager, int32_t Wins);
void HandleRegularSeasonAmountChanged(class UAchievementManager_TA* Manager, int32_t Amount);
```

```

void HandleNewMVPScore(class UAchievementManager_TA* Manager, int32_t Score);
void HandleGameEventAmountWon(class UAchievementManager_TA* Manager, int32_t GamesWon);
void HandleShotAmountBlocked(class UAchievementManager_TA* Manager, int32_t bLocked);
void HandleGoalAmountScored(class UAchievementManager_TA* Manager, int32_t Score);
void HandleDistanceDriven(class UAchievementManager_TA* Manager, float Distance);
void PostActivityFeedSingleString(uint8_t ActivityFeedID, class FString ReplaceString);
void PostActivityFeedMessage(uint8_t ActivityFeedID, TArray<class FString> StringReplaceList);
void eventConstruct();
};

// Class TAGame.ActorFactorySkeletalMeshactorMat_Products_TA
// 0x0000 (0x00B8 - 0x00B8)
class UActorFactorySkeletalMeshactorMat_Products_TA : public
UActorFactorySkeletalMeshMAT
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ActorFactorySkeletalMeshactorMat_Products_TA");
}

return uClassPointer;
};

void eventPostCreateActor(class AActor* NewActor, class USeqAct_ActorFactory*
ActorFactoryData);
};

// Class TAGame.ActorTarget_TA
// 0x0008 (0x0268 - 0x0270)
class AActorTarget_TA : public AActor
{
public:
class UClass* TargetClass; // 0x0268 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ActorTarget_TA");
}
}

```

```

return uClassPointer;
};

void PostBeginPlay();
};

// Class TAGame.AdManager_TA
// 0x0060 (0x0060 - 0x00C0)
class UAdManager_TA : public UObject
{
public:
    uint64_t CacheDuration; // 0x0060 (0x0008)
[0x0000000000000000] TArray<struct FCachedAdImageData> CachedAdImages; // 0x0068
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FBillboardData> BillboardZoneMapping; // 0x0078
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FBillboardUpdateRequestData> BillboardMeshUpdateRequests; // 0x0088
0x0088 (0x0010) [0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
    class UOnlineImageDownloaderWeb* ImageDownloader; // 0x0098
(0x0008) [0x0000000000002000] (CPF_Transient)
    int32_t CurrentAdIndex; // 0x00A0 (0x0004)
[0x0000000000002000] (CPF_Transient)
    class UAdManagerConfig_TA* AdConfig; // 0x00A8 (0x0008)
[0x0000800000000000]
    TArray<struct FAdInfo> Ads; // 0x00B0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AdManager_TA");
        }
    }

    return uClassPointer;
};

void AddBillboardMeshRequest(class UStaticMeshComponent* Mesh);
void HandlePsyNetConnected();
void BatchBillboardRequestsTimer();
void RequestBillboardMeshUpdate(class UStaticMeshComponent* Mesh);
void CompleteBillboardMeshRequests();
class UTexture2DDynamic* GetCachedAdForZoneID(int32_t ZoneID);
void HandleOnlineImageDownloaded(class UTexture2DDynamic* Texture, class UError* Error);
void GetNextImage();
void HandleGetImageURLResponse(class URPC_AdsGet_TA* RPC);
void HandleAdsFromConfig(int32_t MapIndex);
void ClearAllRequests(class FString MapName);
void ClearExpiredImages();

```

```
void eventConstruct();
};

// Class TAGame.RPC_AdsGet_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_AdsGet_TA : public URPC_X
{
public:
    class FString Language; // 0x00E8 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FAdInfo> Ads; // 0x00F8 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_AdsGet_TA");
        }

        return uClassPointer;
    }

    class URPC_AdsGet_TA* SetLanguage(class FString InLanguage);
};

// Class TAGame.AdManagerConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UAdManagerConfig_TA : public UOnlineConfig_X
{
public:
    TArray<struct FMapAds> Ads; // 0x0078 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AdManagerConfig_TA");
        }

        return uClassPointer;
    }

};

// Class TAGame.AgeGate_TA
// 0x007C (0x0064 - 0x00E0)
```

```

class UAgeGate_TA : public UStateObject_X
{
public:
    class UError*           AgeGateError;          // 0x0068 (0x0008)
    [0x0000000000000000]
    class UEpicLogin_X*     EpicLogin;            // 0x0070 (0x0008)
    [0x0000800000000000]
    class UEpicConfig_X*    EpicConfig;           // 0x0078 (0x0008)
    [0x00080000000001] (CPF_Edit)
    struct FScriptDelegate   __SubmitDOBResponse__Delegate; // 0x0080
    (0x0018) [0x00000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate   __QueryAgeGateCallback__Delegate; // 0x0098
    (0x0018) [0x00000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate   __QueryAgeGate__Delegate;        // 0x00B0
    (0x0018) [0x00000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate   __SubmitDobToEos__Delegate;      // 0x00C8
    (0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AgeGate_TA");
        }

        return uClassPointer;
    }

    void HandleQueryAgeGateCompleted(class UAgeGateRequiredResponse* ResponseAsObject,
                                     class UError* InError);
    void HandleDobSubmitted(class UCreateAccountResponse* ResponseAsObject, class UError* Error);
    void HandleCreateAccount(class UCreateAccountResponse* ResponseAsObject, class UError* Error);
    bool ValidateDOB(int32_t Year, int32_t Month, int32_t Day);
    bool IsAgeGateComplete();
    bool IsAgeGateInProgress();
    void HandleEpicLoginSucceeded();
    void NotifyAgeGateRequired();
    void SubmitDOB(int32_t Year, int32_t Month, int32_t Day);
    void RetrieveAgeGateRequired();
    void HandleEpicLoginInjected();
    void eventConstruct();
    void SubmitDobToEos(class FString DOB, class FString Continuation, struct FScriptDelegate Callback);
    void QueryAgeGate(struct FScriptDelegate Callback);
    void QueryAgeGateCallback(class UAgeGateRequiredResponse* ResponseAsObject, class UError* Error);
    void SubmitDOBResponse(class UCreateAccountResponse* ResponseAsObject, class UError* Error);
};

```

```
// Class TAGame.CreateAccountResponse
// 0x0010 (0x0060 - 0x0070)
class UCreateAccountResponse : public UObject
{
public:
class FString Continuation; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CreateAccountResponse");
}

return uClassPointer;
};

};

// Class TAGame.EmptyResponse
// 0x0000 (0x0060 - 0x0060)
class UEmptyResponse : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EmptyResponse");
}

return uClassPointer;
};

};

// Class TAGame.RumblePickups_TA
// 0x00B0 (0x0268 - 0x0318)
class ARumblePickups_TA : public AActor
{
public:
struct FPickupInfo_TA PickupInfo; // 0x0268 (0x0020)
[0x000800A000002020] (CPF_Net | CPF_Transient)
class ASpecialPickup_TA* AttachedPickup; // 0x0288 (0x0008)
[0x0008000000002000] (CPF_Transient)
```

```

int32_t ConcurrentItemCount; // 0x0290 (0x0004)
[0x0008000000002020] (CPF_Net | CPF_Transient)
int32_t SelectedIndex; // 0x0294 (0x0004)
[0x0008000000002000] (CPF_Transient)
int32_t PreviewTimeSeconds; // 0x0298 (0x0004)
[0x0008000000002020] (CPF_Net | CPF_Transient)
struct FScriptDelegate __PickupInfo_ChangeNotify; // 0x02A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __AttachedPickup_ChangeNotify; // 0x02B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __ConcurrentItemCount_ChangeNotify; // 0x02D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __SelectedIndex_ChangeNotify; // 0x02E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __PreviewTimeSeconds_ChangeNotify; // 0x0300
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RumblePickups_TA");
}

return uClassPointer;
};

void __PreviewTimeSeconds_ChangeNotifyFunc();
void __SelectedIndex_ChangeNotifyFunc();
void __ConcurrentItemCount_ChangeNotifyFunc();
void __AttachedPickup_ChangeNotifyFunc();
void __PickupInfo_ChangeNotifyFunc();
int32_t GetNextPickupIndex();
bool CanSelectPickupIndex(int32_t Index);
void NextPickup();
void UpdateAttachedPickup();
void SetPreviewingPickups(unsigned long bPreview);
void SetAvailablePickup(class ASpecialPickup_TA* Pickup, int32_t Index);
void OnPickupUsed(class ACar_TA* Car, class ASpecialPickup_TA* UsedPickup);
void SetConcurrentItemCount(int32_t NewCount);
bool CanUsePickup(class ASpecialPickup_TA* Pickup);
void eventDestroyed();
bool IsAnyPickupAvailable();
void eventPostBeginPlay();
static class ARumblePickups_TA* CreateInstance(class AWorldInfo* InWorldInfo, class
ACar_TA* OwnerCar);
};

// Class TAGame.AIController_Breakout_TA
// 0x0010 (0x07C0 - 0x07D0)
class AAIController_Breakout_TA : public AAIController_Soccar_TA

```

```

{
public:
struct FVector           ClosestEnemyGoalLocation;          // 0x07C0 (0x000C)
[0x0000000000000000]
int32_t                  DefaultSoccarSize;             // 0x07CC (0x0004)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AIController_Breakout_TA");
}

return uClassPointer;
};

void GetGoalProxy(class UBTProxyType* Type, struct FAICachedGoalInfo GoalInfo, struct
FAIProxyData& OutData);
void GetBallProxy(struct FAIProxyData& OutData);
void Init(class AGameEvent_TA* InGameEvent);
};

// Class TAGame.AIProxy_TA
// 0x005C (0x0060 - 0x00BC)
class UAIProxy_TA : public UObject
{
public:
class AActor*              Actor;                      // 0x0060 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FVector               Location;                 // 0x0068 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FVector               Velocity;                // 0x0074 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FRotator              Rotation;                // 0x0080 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FVector               Forward;                 // 0x008C (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FVector               Right;                   // 0x0098 (0x000C)
[0x0000004000002000] (CPF_Transient)
struct FVector               Up;                      // 0x00A4 (0x000C)
[0x0000004000002000] (CPF_Transient)
float                       Speed;                   // 0x00B0 (0x0004)
[0x0000004000002000] (CPF_Transient)
float                       Speed2D;                 // 0x00B4 (0x0004)
[0x0000004000002000] (CPF_Transient)
float                       ForwardSpeed;           // 0x00B8 (0x0004)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AIProxy_TA");
}

return uClassPointer;
};

void Update();
void InitProxy(class AAIManager_TA* AIManager, class AActor* InActor);
};

// Class TAGame.AnimNotify_PlayAKSound_TA
// 0x000C (0x0068 - 0x0074)
class UAnimNotify_PlayAKSound_TA : public UAnimNotify_Scripted
{
public:
class UAkSoundCue*           SFX_SoundCue;           // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                 bPlayInReverse : 1;    // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNotify_PlayAKSound_TA");
}

return uClassPointer;
};

void eventNotify(class AActor* Owner, class UAnimNodeSequence* AnimSeqInstigator);

// Class TAGame.AnimNotify_SetBoneVisibility_TA
// 0x0018 (0x0068 - 0x0080)
class UAnimNotify_SetBoneVisibility_TA : public UAnimNotify_Scripted
{
public:
struct FName                  SetVisibleName;        // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                 bSetVisible : 1;      // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
uint8_t                       SetVisibilityType;    // 0x0074 (0x0001)
[0x0000000000000001] (CPF_Edit)
class USkeletalMeshComponent* SKMComp;            // 0x0078
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNotify_SetBoneVisibility_TA");
}

return uClassPointer;
};

void SetBoneVisibility();
void SetSocketVisibility();
void eventNotify(class AActor* Owner, class UAnimNodeSequence* AnimSeqInstigator);
};

// Class TAGame.AnimNotify_SetFXState_TA
// 0x0008 (0x0068 - 0x0070)
class UAnimNotify_SetFXState_TA : public UAnimNotify_Scripted
{
public:
class UFXActorEvent_X*           FXEvent;                      // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNotify_SetFXState_TA");
}

return uClassPointer;
};

void eventNotify(class AActor* Owner, class UAnimNodeSequence* AnimSeqInstigator);
};

// Class TAGame.AnimNotify_TriggerTurntableEvent_TA
// 0x000C (0x0068 - 0x0074)
class UAnimNotify_TriggerTurntableEvent_TA : public UAnimNotify_Scripted
{
public:
struct FName                  TurntableEventName;          // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long                 bPlayInReverse : 1;        // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AnimNotify_TurntableEvent_TA");
}

return uClassPointer;
};

void eventNotify(class AActor* Owner, class UAnimNodeSequence* AnimSeqInstigator);
};

// Class TAGame.TurnTableActor_TA
// 0x008C (0x02CC - 0x0358)
class ATurnTableActor_TA : public ASkeletalMeshActor
{
public:
class URotateComponent_TA*           RotateComponent;           // 0x02D0
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
unsigned long                      bIsSwapping : 1;          // 0x02D8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long                      bInReveal : 1;           // 0x02D8 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long                      bCarLocked : 1;          // 0x02D8 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
unsigned long                      bMultiDrop : 1;           // 0x02D8 (0x0004)
[0x0000000000002000] [0x00000008] (CPF_Transient)
struct FName                       SwapCarName;             // 0x02DC (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                       LiftItemName;            // 0x02E4 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                       LockCarName;             // 0x02EC (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                       TriggerRevealCompleteName; // 0x02F4 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t                           DelayedPreviewSlot;        // 0x02FC (0x0004)
[0x0000000000002000] (CPF_Transient)
float                             TurntableRiseDelay;        // 0x0300 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UOnlineProduct_TA*          RevealProduct;             // 0x0308 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate            __EventCarSwapComplete__Delegate; // 0x0310
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate            __EventRevealStarted__Delegate; // 0x0328
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate            __EventRevealComplete__Delegate; // 0x0340
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TurnTableActor_TA");
}

return uClassPointer;
};

void EnableCollision(unsigned long bEnable);
void StopMovement();
struct FRotator GetDiscRotation();
void SetBoneRotation(struct FRotator NewRot);
void HandleRotationChanged(class URotateComponent_TA* InRotateComponent, struct
FRotator NewRotation);
void SetDiscRotation(float InDirectionMag, float DeltaTime);
bool CanSubstituteCars();
void LockCar();
void eventOnAnimEnd(class UAnimNodeSequence* SeqNode, float PlayedTime, float
ExcessTime);
void TriggerRevealComplete();
void DelayPreviewSlot(int32_t SlotIndex);
void OnStartReveal();
void DoDelayedReveal();
void DoReveal(class UOnlineProduct_TA* OnlineProduct, unsigned long bMultiProductDrop);
void TriggerNamedEvent(struct FName NamedEvent);
void SwapCars(unsigned long bReverse);
void eventDestroyed();
void eventPostBeginPlay();
void EventRevealComplete(int32_t SlotIndex);
void EventRevealStarted();
void EventCarSwapComplete();
};

// Class TAGame.AntiAddiction_TA
// 0x0004 (0x0060 - 0x0064)
class UAntiAddiction_TA : public UObject
{
public:
int32_t Hours; // 0x0060 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AntiAddiction_TA");
}
}

```

```

return uClassPointer;
};

void HandleHourReached();
void HandleCountry(class FString Country);
void eventConstruct();
};

// Class TAGame.Notification_AntiAddiction_TA
// 0x0000 (0x0170 - 0x0170)
class UNotification_AntiAddiction_TA : public UNotification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Notification_AntiAddiction_TA");
}

return uClassPointer;
};

};

// Class TAGame.ApplyVelocityHitHandler_TA
// 0x000C (0x0150 - 0x015C)
class UApplyVelocityHitHandler_TA : public UExplosionHitHandler_TA
{
public:
struct FVector VelocityToApply; // 0x0150 (0x000C)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ApplyVelocityHitHandler_TA");
}

return uClassPointer;
};

void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

```

```

// Class TAGame.AppMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UAppMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AppMetrics_TA");
}

return uClassPointer;
};

void AppStart();
};

// Class TAGame.GameplayMusicPlayer_TA
// 0x0080 (0x0060 - 0x00E0)
class UGameplayMusicPlayer_TA : public UObject
{
public:
class UMusicPlayerSave_TA*           MusicSave;           // 0x0060 (0x0008)
[0x0000000000002000] (CPF_Transient)
class AGameEvent_TA*                 CurrentGameEvent; // 0x0068 (0x0008)
[0x0000800000000000]
unsigned long                        bMuted : 1;       // 0x0070 (0x0004)
[0x0000000000000000] [0x00000001]
struct FName                         LoadingScreenState; // 0x0074 (0x0008)
[0x0000000000000000]
struct FName                         CurrentPlaylist; // 0x007C (0x0008)
[0x0000000000000000]
uint8_t                             MusicPlayerState; // 0x0084 (0x0001)
[0x0000000000000000]
struct FName                         LoadingScreenStateGroup; // 0x0088 (0x0008)
[0x0000000000000000]
struct FName                         LoadingScreenState_MusicEnabled; // 0x0090
(0x0008) [0x0000000000000000]
struct FName                         LoadingScreenState_MusicDisabled; // 0x0098
(0x0008) [0x0000000000000000]
struct FName                         LoadingScreenState_None; // 0x00A0 (0x0008)
[0x0000000000000000]
struct FName                         GameplayMusicStateGroup; // 0x00A8 (0x0008)
[0x0000000000000000]
struct FName                         GameplayMusicState_Enabled; // 0x00B0
(0x0008) [0x0000000000000000]
struct FName                         GameplayMusicState_Disabled; // 0x00B8
(0x0008) [0x0000000000000000]
struct FName                         GameplayMusicState_None; // 0x00C0 (0x0008)

```

```

[0x0000000000000000]
struct FScriptDelegate           __EventStateChanged__Delegate;          // 0x00C8
(0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameplayMusicPlayer_TA");
}

return uClassPointer;
};

void HandleEventDestroyed(class AGameEvent_TA* GameEvent);
void HandleMatchTypeSet(class AGameEvent_TA* GameEvent);
void HandleLoadingMovieClosed();
void HandleGameEventChanged();
void HandlePrimaryPlayerChanged(class UOnlineGameAccount_X* InAccount, struct
FUniqueNetId PlayerID);
void SetLoadingScreenState(struct FName InState);
struct FName GetLoadingScreenState(unsigned long blsLoadingMainMenu);
void UpdateLoadingScreenState(class FString MapName);
void UpdateBGM();
void BroadcastStateChange();
void SetState(uint8_t InState);
bool IsMainMenu();
bool IsTrainingMatch(class UMatchType_TA* MatchType);
bool IsReplay();
bool IsActiveForMatchType(class UMatchType_TA* MatchType);
uint8_t GetPlaylistType(class UGameSettingPlaylist_X* Playlist);
void SetMuted(unsigned long bInMuted);
void SetMusicSave(class UMusicPlayerSave_TA* InMusicSave);
bool IsPlayingMusic();
void RegisterLoadingScreenCallbacks();
void eventConstruct();
void EventStateChanged(class UGameplayMusicPlayer_TA* GameplayMusicPlayer, uint8_t
NewState);
};

// Class TAGame.MusicPlayerSave_TA
// 0x0060 (0x00C8 - 0x0128)
class UMusicPlayerSave_TA : public USaveObject_TA
{
public:
TArray<struct FPlaylistState>           PlaylistsUpdate22;          // 0x00C8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
uint8_t                      GameplayMusicSetting;          // 0x00D8 (0x0001)
[0x0000000000000000]
unsigned long                  bPlayMusicInMenu : 1;          // 0x00DC (0x0004)
[0x0008000000000000] [0x00000001]

```

```

unsigned long          bPlayMusicInTraining : 1;           // 0x00DC (0x0004)
[0x0008000000000000] [0x00000002]
unsigned long          bPlayMusicInGame : 1;            // 0x00DC (0x0004)
[0x0008000000000000] [0x00000004]
struct FScriptDelegate      __bPlayMusicInMenu__ChangeNotify;    // 0x00E0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __bPlayMusicInTraining__ChangeNotify;   // 0x00F8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __bPlayMusicInGame__ChangeNotify;     // 0x0110
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicPlayerSave_TA");
}

return uClassPointer;
};

void __MusicPlayerSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __bPlayMusicInGame__ChangeNotifyFunc();
void __bPlayMusicInTraining__ChangeNotifyFunc();
void __bPlayMusicInMenu__ChangeNotifyFunc();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
bool IsModified();
TArray<struct FPlaylistState> GetPlaylists();
void SetSelected(struct FName Playlist, unsigned long bEnabled);
void OnCreate();
};

// Class TAGame.MatchType_Custom_TA
// 0x000C (0x009C - 0x00A8)
class UMatchType_Custom_TA : public UMatchType_TA
{
public:
unsigned long          bOnePlayerSelectedTeam : 1;           // 0x00A0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
int32_t                LargestMaxTeamSize;                 // 0x00A4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Custom_TA");
}
}

```

```

return uClassPointer;
};

struct FUniqueNetId __MatchType_Custom_TA__ShouldStartMatch_0x1(class APRI_TA* PRI);
bool CanCheckForBannedPlayers();
bool CanIncreaseMaxTeamSize();
bool UseCustomMatchSettings();
bool ShouldSetGameOwner();
bool AllowMatchAdmin();
bool AllowSpectators();
bool RecordPlayedMap();
bool ShouldShowDisconnectedPlayersOnScoreboard();
class ATeam_TA* PickTeam(class AController* C);
bool CanRestartPlayer(class AController* NewPlayer);
bool ShouldAutoSelectTeam(class AController* NewPlayer);
int32_t GetMaxTeamSize();
bool ChooseTeam(int32_t TeamIndex, class APlayerController_TA* Player);
bool CanJoinATeam();
void OnPlayerAddedToTeam();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
void SetGameData();
void InitMaxTeamSize();
void OnInitGameEvent();
};

// Class TAGame.MatchType_Offline_TA
// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_Offline_TA : public UMatchType_Custom_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Offline_TA");
}

return uClassPointer;
};

struct FName GetOfflinePlaylistName(class AGameEvent_TA* InGameEvent, class FString Options);
void OnInitGameEvent();
void Init(class AGameEvent_TA* InGameEvent, class FString Options);
};

// Class TAGame.MatchType_Tutorial_TA
// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_Tutorial_TA : public UMatchType_Offline_TA

```

```

{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Tutorial_TA");
}

return uClassPointer;
};

bool AllowMatchAdmin();
void InitCustomMatchSettings(struct FCustomMatchSettings Settings);
struct FName GetOfflinePlaylistName(class AGameEvent_TA* InGameEvent, class FString Options);
int32_t GetMaxTeamSize();
};

// Class TAGame.StreamerSafeConfig_TA
// 0x0040 (0x0078 - 0x00B8)
class UStreamerSafeConfig_TA : public UOnlineConfig_X
{
public:
TArray<class FString> StreamerSafeSoundCueNames; // 0x0078
(0x0010) [0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FPlaylistTrack> FallbackStreamerSafeTitleTracks; // 0x0088
(0x0010) [0x0001004000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bPlayerSettingEnabled : 1; // 0x0098 (0x0004)
[0x0009000000002000] [0x00000001] (CPF_Transient)
struct FScriptDelegate __bPlayerSettingEnabled__ChangeNotify; // 0x00A0
(0x0018) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StreamerSafeConfig_TA");
}

return uClassPointer;
};

void __bPlayerSettingEnabled__ChangeNotifyFunc();
bool IsStreamerSafe(class UAkSoundCue* SoundCue);
};

```

```
// Class TAGame.ProductAsset_MusicStingers_TA
// 0x001C (0x00F8100 - 0x0114C)
class UProductAsset_MusicStingers_TA : public UProductAsset_TA
{
public:
    class UAkSoundCue*           GoalScoredStinger;          // 0x00F8100
    [0x0008] [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           EpicSaveStinger;          // 0x01008 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue*           MVPAnthem;                // 0x01108 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                 bIsMusic : 1;            // 0x01108 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_MusicStingers_TA");
        }

        return uClassPointer;
    };

};

// Class TAGame.ArenaSoundPlayer_TA
// 0x0008 (0x0268 - 0x0270)
class AArenaSoundPlayer_TA : public AActor
{
public:
    class UAkSoundSource*         SoundSource;               // 0x0268 (0x0008)
    [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ArenaSoundPlayer_TA");
        }

        return uClassPointer;
    };

    void eventPostBeginPlay();
};

// Class TAGame.MapInfo_TA
```

```

// 0x00B8 (0x0060 - 0x0118)
class UMapInfo_TA : public UMapInfo
{
public:
    struct FWorldColorData           WorldColors;          // 0x0060 (0x0040)
    [0x0000000000000001] (CPF_Edit)
    uint8_t                          MirrorFieldType;      // 0x00A0 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    int32_t                          MirrorFieldXYRotation; // 0x00A4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                    bCanMirrorTraining : 1; // 0x00A8 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long                    bMirrorTrainingSwapsGoals : 1; // 0x00A8 (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    unsigned long                    bPlayGlobalCrowdSounds : 1; // 0x00A8 (0x0004)
    [0x0000000000000001] [0x00000004] (CPF_Edit)
    struct FVector                  FieldCenter;         // 0x00AC (0x000C)
    [0x0000000000000001] (CPF_Edit)
    float                           MinStadiumBrightness; // 0x00B8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FRotator                MapResetStartRotation; // 0x00BC (0x000C)
    [0x0000000000000001] (CPF_Edit)
    float                           MapResetStartHeight; // 0x00C8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           MapResetStartDistance; // 0x00CC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           MaxWaitTimeTillBallHit; // 0x00D0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           GoalScoredDistance; // 0x00D4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           MaxGoalYZoomOutDistance; // 0x00D8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           MaxGoalXZoomOutDistance; // 0x00DC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           CountDownFOV;          // 0x00E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           BackAwayFromWallDistance; // 0x00E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FVector                  PylonExtentOverride; // 0x00E8 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    class UArenaSoundSettings_TA*   ArenaSoundSettings; // 0x00F8
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class UCrowdSoundsBase_TA*      CrowdSounds;        // 0x0100
    (0x0008) [0x0010000000000001] (CPF_Edit)
    TArray<class ACrowdActor_TA*> CrowdActors;        // 0x0108
    (0x0010) [0x0000000004000001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {

```

```

uClassPointer = UObject::FindClass("Class TAGame.MapInfo_TA");
}

return uClassPointer;
};

};

// Class TAGame.SoundSettingsSave_TA
// 0x0160 (0x00C8 - 0x0228)
class USoundSettingsSave_TA : public USaveObject_TA
{
public:
float [0x0008000000000000] MasterVolume; // 0x00C8 (0x0004)
float [0x0008000000000000] SoundVolume; // 0x00CC (0x0004)
float [0x0008000000000000] MusicVolume; // 0x00D0 (0x0004)
float [0x0008000000000000] GameplayMusicVolume; // 0x00D4 (0x0004)
float [0x0008000000000000] AmbientVolume; // 0x00D8 (0x0004)
float [0x0008000000000000] CrowdVolume; // 0x00DC (0x0004)
float [0x0008000000000000] VoiceVolume; // 0x00E0 (0x0004)

unsigned long [0x0008000000000000] bMuteOnFocusLost : 1; // 0x00E4 (0x0004)
unsigned long [0x0008000000000000] bStreamerSafeAudioEnabled : 1; // 0x00E4 (0x0004)
unsigned long [0x0008000000000000] blsFinishedLoadingSaveSettings : 1; // 0x00E4 (0x0004)
uint8_t [0x0008000000000000] OutputType; // 0x00E8 (0x0001)
uint8_t [0x0009000000000000] DynamicRangeType; // 0x00E9 (0x0001)
uint8_t [0x0008000000000000] MusicStingersSetting; // 0x00EA (0x0001)

struct FScriptDelegate _MasterVolume_ChangeNotify; // 0x00F0 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _SoundVolume_ChangeNotify; // 0x0108 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _MusicVolume_ChangeNotify; // 0x0120 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _GameplayMusicVolume_ChangeNotify; // 0x0138 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _AmbientVolume_ChangeNotify; // 0x0150 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _CrowdVolume_ChangeNotify; // 0x0168 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _VoiceVolume_ChangeNotify; // 0x0180 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _bMuteOnFocusLost_ChangeNotify; // 0x0198

```

```

(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bStreamerSafeAudioEnabled__ChangeNotify; // 0x01B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __bIsFinishedLoadingSaveSettings__ChangeNotify; // 0x01C8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __OutputType__ChangeNotify; // 0x01E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __DynamicRangeType__ChangeNotify; // 0x01F8
(0x0018) [0x0001000004000000] (CPF_NeedCtorLink)
struct FScriptDelegate           __MusicStingersSetting__ChangeNotify; // 0x0210
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SoundSettingsSave_TA");
}

return uClassPointer;
};

void __MusicStingersSetting__ChangeNotifyFunc();
void __DynamicRangeType__ChangeNotifyFunc();
void __OutputType__ChangeNotifyFunc();
void __bIsFinishedLoadingSaveSettings__ChangeNotifyFunc();
void __bStreamerSafeAudioEnabled__ChangeNotifyFunc();
void __bMuteOnFocusLost__ChangeNotifyFunc();
void __VoiceVolume__ChangeNotifyFunc();
void __CrowdVolume__ChangeNotifyFunc();
void __AmbientVolume__ChangeNotifyFunc();
void __GameplayMusicVolume__ChangeNotifyFunc();
void __MusicVolume__ChangeNotifyFunc();
void __SoundVolume__ChangeNotifyFunc();
void __MasterVolume__ChangeNotifyFunc();
};

// Class TAGame.SoundSettingsManager_TA
// 0x0018 (0x0060 - 0x0078)
class USoundSettingsManager_TA : public UObject
{
public:
class USoundSettingsSave_TA*           SoundSave; // 0x0060 (0x0008)
[0x0000004000002000] (CPF_Transient)
class USoundSettingsControllerSave_TA*   SoundSaveController; // 0x0068
(0x0008) [0x0000004000002000] (CPF_Transient)
class ULocalPlayer_TA*                 PrimaryPlayer; // 0x0070 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()

```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SoundSettingsManager_TA");  
}  
  
return uClassPointer;  
};  
  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0xb();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0xa();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x9();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x8();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x7();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x6();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x5();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x4();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x3();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x2();  
void __SoundSettingsManager_TA__HandleSoundSettingsLoaded_0x1();  
void __SoundSettingsManager_TA__HandleSoundSettingsControllerLoaded_0x2();  
void __SoundSettingsManager_TA__HandleSoundSettingsControllerLoaded_0x1();  
void SetStreamerSafeAudio(unsigned long bStreamerSafeAudioEnabled);  
void SetIsFinishedLoadingSaveSettings(unsigned long bIsFinishedLoadingSaveSettings);  
void SetMuteOnFocusLost(unsigned long bMuteOnFocusLost);  
void SetControllerSoundMode(uint8_t NewValue);  
void HandleLocalPlayerCountChange(class ULocalPlayer* Player);  
void SetControllerVolume(float NewValue);  
void SetDynamicRange(uint8_t NewValue);  
void SetOutputType(uint8_t NewValue);  
float GetMasterVolumeValue();  
void SetVoiceVolume(float NewValue);  
void SetCrowdVolume(float NewValue);  
void SetAmbientVolume(float NewValue);  
void SetGameplayMusicVolume(float NewValue);  
void SetMusicVolume(float NewValue);  
void SetSoundVolume(float NewValue);  
void ApplyControllerSoundSettings();  
void HandleSoundSettingsControllerUnloaded(class USoundSettingsControllerSave_TA* InSoundSave);  
void HandleSoundSettingsControllerLoaded(class USoundSettingsControllerSave_TA* InSoundSave);  
void ApplySoundSettings();  
void HandleSoundSettingsUnloaded(class USoundSettingsSave_TA* InSoundSave);  
void HandleSoundSettingsLoaded(class USoundSettingsSave_TA* InSoundSave);  
void InitFromPlayer(class ULocalPlayer_TA* Player);  
void HandlePrimaryPlayerChange(class ULocalPlayer* Old, class ULocalPlayer_TA* NewPlayer);  
void HandleLocalPlayerLeave(class ULocalPlayer_TA* Player);  
void HandleLocalPlayerJoin(class ULocalPlayer_TA* Player);  
void Init();  
};
```

```

// Class TAGame.StatFactory_TA
// 0x01E0 (0x0278 - 0x0458)
class AStatFactory_TA : public AStatFactoryBase_TA
{
public:
    struct FStatEventCollection           Events;           // 0x0278 (0x0158)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    TArray<struct FBallInfo>           BallCache;        // 0x03D0 (0x0010)
    [0x0000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
    float                             GoalSizeFudge;     // 0x03E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             ShotMaxTime;       // 0x03E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             ShotArriveTime;    // 0x03E8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             ShotMaxDistance;   // 0x03EC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             RedZoneDistance;  // 0x03F0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             JuggleMinSpeed;   // 0x03F4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             BulletHitSpeed;    // 0x03F8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             AerialHitHeight;   // 0x03FC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             AerialHitMinRelSpeed; // 0x0400 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             BicycleSpinSpeed; // 0x0404 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                             LongGoalDistance; // 0x0408 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                           HatTrickThreshold; // 0x040C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                           PlaymakerThreshold; // 0x0410 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                           SaviorThreshold;    // 0x0414 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                           DemolitionThreshold; // 0x0418 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                     bGaveFirstTouch : 1; // 0x041C (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    class AGameEvent_Soccar_TA*      SoccarGame;        // 0x0420
    (0x0008) [0x0000000000002000] (CPF_Transient)
    class APRI_TA*                  ScoringPRI;        // 0x0428 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    TArray<class ACar_TA*>          CarsIssuedAnyFiveHit; // 0x0430 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    struct FScriptDelegate           __EventGaveStat__Delegate; // 0x0440
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatFactory_TA");
}

return uClassPointer;
};

void HandlePlayerRestarted(class AGameEvent_TA* GameEvent, class ACar_TA* PlayerCar);
void RegisterBalls();
void RegisterCars();
void UnregisterCars();
void HandleBumpedCar(class ACar_TA* Car, class ACar_TA* HitCar, struct FVector HitLocation);
uint8_t InitiatorFiveHitType(class ACar_TA* Initiator, class ACar_TA* Receiver);
void OnBallAdded(class AGameEvent_Soccar_TA* GameEventSoccar, class ABall_TA* Ball);
void HandleMatchMVP(class APRI_TA* PRI);
void OnMatchEnded(class AGameEvent_Soccar_TA* SoccarEvent);
void DetermineTouchCounts(class AGameEvent_Soccar_TA* SoccarEvent);
void DeterminePickupCounts(class AGameEvent_Soccar_TA* SoccarEvent);
void OnActiveRoundChanged(class AGameEvent_Soccar_TA* InGameEvent);
void BackflipCars(TArray<class ACar_TA*> CarsToFlip);
void BackflipCar(class ACar_TA* Car);
void OnCarDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void OnGameStateChanged(class AGameEvent_TA* InGame);
void OnGoalScored(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIndex);
void OnCarTouch(class ABall_TA* Ball, class ACar_TA* HitCar, uint8_t HitType);
void OnBallHitWorld(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNormal);
void TickBalls(float DeltaTime);
void Tick(float DeltaTime);
void UpdateRedZone(int32_t BallIdx);
class UGoal_TA* CalculateShotGoal(class UGoal_TA* GoalToTest, class ABall_TA* Ball, float ArriveTime);
void UpdateShotGoal(int32_t BallIdx);
bool IsBallBehindGoal(class ABall_TA* Ball, class UGoal_TA* Goal);
class UGoal_TA* FindShotGoal(class ABall_TA* Ball);
bool IsBulletHit(struct FBallHitInfo Hit);
bool IsBackwardsHit(struct FBallHitInfo Hit);
float GetDeltaSpeed(struct FBallHitInfo Hit);
bool IsTurtleHit(struct FBallHitInfo Hit);
bool IsJuggleHit(struct FBallHitInfo Hit);
bool IsBicycleHit(struct FBallHitInfo Hit);
bool IsLocationFarOffGround(struct FVector InLocation);
bool IsAerialHit(struct FBallHitInfo Hit);
bool IsInRedZone(class UGoal_TA* Goal, class ABall_TA* Ball);
bool IsEpicSave(class ABall_TA* Ball, struct FBallHitInfo Hit, class UGoal_TA* Goal);
float GetHitDistanceToGoal(class ABall_TA* Ball, struct FBallHitInfo Hit, class UGoal_TA* Goal);
void BroadcastStatTickerEvent(class APRI_TA* Receiver, class APRI_TA* Victim, class UStatEvent_TA* StatEvent);
void GiveScore(class APRI_TA* ToPRI, class UStatEvent_TA* StatEvent, class ABall_TA* Ball, int32_t BallHitIndex, class APRI_TA* Victim, class UGoal_TA* Goal, int32_t Count);
void UnregisterCar(class APawn_X* Car);
void RegisterCar(class ACar_TA* Car);

```

```

void UnregisterBall(class APawn_X* Ball);
void RegisterBall(class ABall_TA* Ball);
int32_t GetBallIndex(class ABall_TA* Ball);
void SetGameEvent(class AGameEvent_Soccar_TA* InGameEvent);
void EventGaveStat(class AStatFactory_TA* Factory, class APRI_TA* ToPRI, class
UStatEvent_TA* StatEvent, class ABall_TA* Ball, int32_t BallHitIndex, class APRI_TA* Victim,
class UGoal_TA* Goal);
};

// Class TAGame.MatchType_Freeplay_TA
// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_Freeplay_TA : public UMatchType_Tutorial_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Freeplay_TA");
}

return uClassPointer;
};

};

// Class TAGame.AssetAttribute_ChangeProductDrawScale_TA
// 0x0010 (0x0060 - 0x0070)
class UAssetAttribute_ChangeProductDrawScale_TA : public UProductAssetAttribute_TA
{
public:
TArray<struct FProductSlotNewDrawScale>      NewProductDrawScales;           // 0x0060 (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.AssetAttribute_ChangeProductDrawScale_TA");
}

return uClassPointer;
};

float GetDrawScale(class UProductSlot_TA* SlotToFind);
};

```

```

// Class TAGame.AssetAttribute_ForceWheelAxle_TA
// 0x0010 (0x0060 - 0x0070)
class UAssetAttribute_ForceWheelAxle_TA : public UProductAssetAttribute_TA
{
public:
    uint8_t AxleToForce; // 0x0060 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    class UProductAsset_Wheel_TA* ForcedWheel; // 0x0068
    (0x0008) [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AssetAttribute_ForceWheelAxle_TA");
        }

        return uClassPointer;
    };
}

// Class TAGame.AssetAttribute_GoalCountChanging_TA
// 0x0030 (0x0060 - 0x0090)
class UAssetAttribute_GoalCountChanging_TA : public UProductAssetAttribute_TA
{
public:
    TArray<struct FGoalCountAndAsset> Assets; // 0x0060 (0x0010)
    [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    int32_t CurrentGoalCount; // 0x0070 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    int32_t LastIndexUsed; // 0x0074 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    unsigned long bIsGaragePreview : 1; // 0x0078 (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    class UObject* CurrentObjectToUse; // 0x0080 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class UProductAsset_TA* AssetBeingUsed; // 0x0088 (0x0008)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.AssetAttribute_GoalCountChanging_TA");
        }
    }
}

```

```

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
int32_t GetMaxGoalCount();
bool IsInReplayPlayback();
class UObject* GetAssetFromProductAsset();
void GarageFXActorSpawned(class USeqAct_SpawnFXActor_TA* SpawnedFXActor);
void CheckAndChangeLastIndexUsed(int32_t IndexToCheck);
void CheckAndIncrementGoalCount();
class UObject* GetNextAsset(class UProductAsset_TA* Asset);
void GarageClosed();
void SetupForGarage(class UProductAsset_TA* Asset);
};

// Class TAGame.AssetAttribute_PostMatchCelebration_TA
// 0x0030 (0x0060 - 0x0090)
class UAssetAttribute_PostMatchCelebration_TA : public UProductAssetAttribute_TA
{
public:
class USkeletalMesh* PostMatchCelebrationMesh; // 0x0060
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FName MVPPostMatchCelebrationAnim; // 0x0068
(0x0008) [0x0000000000000001] (CPF_Edit)
TArray<struct FName> RegularPostMatchCelebrationAnims; // 0x0070
(0x0010) [0x0000000000040001] (CPF_Edit | CPF_NeedCtorLink)
class USkeletalMeshComponent_SplitBody_TA* SplitMeshComp; // 0x0080 (0x0008) [0x0000004004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class USkeletalMeshComponent_SplitBody_TA* InstancedSplitMeshComp; // 0x0088 (0x0008) [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AssetAttribute_PostMatchCelebration_TA");
}

return uClassPointer;
};

uint8_t eventCheckIfErrorOrWarning(class FString& OutString);
class UProductAttribute_TA* eventGenerateProductAttribute(class UObject* InParent);
bool eventRequiresGeneratedProductAttribute();
void StopAnim(class UCarMeshComponent_TA* CarMesh);
void ResetMeshComponent(class UCarMeshComponent_TA* CarMesh);
void CreateCarMeshAnimTree(class UCarMeshComponent_TA* CarMesh);
void SetPostMatchMeshOntoMeshComponent(class UCarMeshComponent_TA* CarMesh);
float PlayPostMatchAnim(class UCarMeshComponent_TA* CarMesh, struct FName

```

```

PostMatchAnim);
int32_t GetRegularPostMatchAnimIdx(struct FName AnimName);
TArray<struct FName> GetRegularPostMatchAnims();
struct FName GetMVPPostMatchAnim();
struct FName GetPostMatchMeshName();
bool HasPMCMesh();
};

// Class TAGame.AssetAttribute_SoundOverrides_TA
// 0x0010 (0x0060 - 0x0070)
class UAssetAttribute_SoundOverrides_TA : public UProductAssetAttribute_TA
{
public:
TArray<struct FFXAttachmentSoundOverride>      SoundOverrides;          // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AssetAttribute_SoundOverrides_TA");
}

return uClassPointer;
};

};

// Class TAGame.AttachmentAnim_TA
// 0x0018 (0x0074 - 0x008C)
class UAttachmentAnim_TA : public UAttachmentBehavior_TA
{
public:
class UAnimSet*           AnimSet;          // 0x0078 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName               AnimName;         // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)
unsigned long              bIgnoreAnimationForThumbnails : 1;    // 0x0088
(0x0004) [0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AttachmentAnim_TA");
}

return uClassPointer;
}

```

```

};

static bool IsComponentInThumbnailScene(class UComponent* InComponent);
void OnInit();
};

// Class TAGame.AudioFieldSideComponent_TA
// 0x0024 (0x00A4 - 0x00C8)
class UAudioFieldSideComponent_TA : public UActorComponent_X
{
public:
struct FName           FieldSideRTPC;          // 0x00A8 (0x0008)
[0x0000000000000000]
float                  OneOverFieldLength;    // 0x00B0 (0x0004)
[0x0000000000000000]
class AGameEvent_Soccar_TA*   GameEvent;        // 0x00B8
(0x0008) [0x0000000000000000]
class UArenaReflectionsManager_TA* ArenaReflectionsManager; // 0x00C0
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioFieldSideComponent_TA");
}

return uClassPointer;
};

void HandleGameEventSet(class AGameEvent_Soccar_TA* NewGameEvent);
void HandleReflectionsManagerSet(class UArenaReflectionsManager_TA*
NewReflectionsManager);
void UpdateFieldSideParams();
void UpdateCanTick();
void eventTick(float DeltaTime);
void eventDetached();
void eventAttached();
};

// Class TAGame.AudioMixStateComponent_TA
// 0x009C (0x00A4 - 0x0140)
class UAudioMixStateComponent_TA : public UActorComponent_X
{
public:
struct FName           CameraBallFocusStateGroup; // 0x00A8 (0x0008)
[0x0000000000000000]
struct FName           CameraBallFocus_On;       // 0x00B0 (0x0008)
[0x0000000000000000]
struct FName           CameraBallFocus_Off;     // 0x00B8 (0x0008)
[0x0000000000000000]

```

```

struct FName [0x0000000000000000] GameState; // 0x00C0 (0x0008)
struct FName [0x0000000000000000] PlayerCountStateGroup; // 0x00C8 (0x0008)
struct FName [0x0000000000000000] PlayerCount_4s; // 0x00D0 (0x0008)
struct FName [0x0000000000000000] PlayerCount_3s; // 0x00D8 (0x0008)
struct FName [0x0000000000000000] PlayerCount_2s; // 0x00E0 (0x0008)
struct FName [0x0000000000000000] PlayerCount_1s; // 0x00E8 (0x0008)
struct FName [0x0000000000000000] PlayerCount_None; // 0x00F0 (0x0008)
struct FName [0x0000000000000000] MajorityFarStateGroup; // 0x00F8 (0x0008)
struct FName [0x0000000000000000] MajorityFar_True; // 0x0100 (0x0008)
struct FName [0x0000000000000000] MajorityFar_None; // 0x0108 (0x0008)
struct FName [0x0000000000000000] MajorityFarRPC; // 0x0110 (0x0008)
struct FName [0x0000000000000000] IsTeamRPC; // 0x0118 (0x0008)
struct FName [0x0000000000000000] FocusCarRPC; // 0x0120 (0x0008)
struct FName (0x0008) [0x0000000000000000] LiveReplay_InCameraViewRPC; // 0x0128
int32_t [0x0000000000000000] NumLocalPlayers; // 0x0130 (0x0004)
unsigned long [0x0000000000000000] [0x00000001] bPlayerUsingBallCam : 1; // 0x0134 (0x0004)
unsigned long [0x0000000000000000] [0x00000002] blIsActiveState : 1; // 0x0134 (0x0004)
unsigned long [0x0000000000000000] [0x00000004] bSpectating : 1; // 0x0134 (0x0004)
unsigned long [0x0000000000000000] [0x00000008] blIsMajorityFar : 1; // 0x0134 (0x0004)
unsigned long [0x0000000000000000] [0x00000010] blInLiveReplay : 1; // 0x0134 (0x0004)
float [0x0000000000000000] MajorityFarMinDistanceSq; // 0x0138 (0x0004)
float [0x0000000000000000] BallListenerOffset; // 0x013C (0x0004)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioMixStateComponent_TA");
}

```

```

return uClassPointer;
};

void SetPlayerCountState(int32_t PlayerCount);
void UpdateNumLocalPlayers(class UGameViewportClient_TA* GVC, int32_t LocalPlayerCount);
void BallCamToggle();
bool IsActiveState(struct FName InGameState);
void HandlePlayerCountChanged(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
bool IsNotSpectator(class APRI_TA* InPRI);
void HandlePawnTypeChanged(class APRI_TA* InPRI);
void HandleFocusCarChanged(class AActor* NewFocusCar);
void HandleGameEventSet(class APRI_TA* NewPRI);
void HandlePersistentCameraSet(class APRI_TA* PRI);
void HandleStateChanged(class AGameEvent_TA* GameEvent);
void HandleBallCamToggle(class ACameraSettingsActor_TA* Cam);
void HandleCarSet(class APRI_TA* InPRI);
void HandlePawnChange(class APlayerController_X* InPC, class APawn* OldPawn, class APawn* NewPawn);
void ClearInCameraViewRTPCs();
void UpdateInCameraViewRTPCs();
void ClearFocusCarRTPCs();
void ClearMajorityFarRTPCs();
void UpdateMajorityFarRTPCs();
void SetMajorityFar(unsigned long bSet);
float GetProportionalCarDistance(class ACar_TA* InCar);
void UpdateMajorityFar();
void UpdateIsTeamRTPCs();
void UpdateBallListenerPosition();
void eventTick(float DeltaTime);
void UpdateTickEnabled();
void eventDetached();
void eventAttached();
};

// Class TAGame.ReplayDirector_TA
// 0x0188 (0x0268 - 0x03F0)
class AReplayDirector_TA : public AActor
{
public:
float SlomoPreScoreTime; // 0x0268 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SlomoPostScoreTime; // 0x026C (0x0004)
[0x0000000000000001] (CPF_Edit)
float SlomoDefendTime; // 0x0270 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SlomoDefendDistance; // 0x0274 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SlomoTimeDilation; // 0x0278 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinReplayTime; // 0x027C (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxReplayTime; // 0x0280 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

float          ReplayPadding;           // 0x0284 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*      ReplayStartSound;        // 0x0288 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*      SlomoStartSound;        // 0x0290 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*      SlomoStopSound;         // 0x0298 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName        LiveReplayStateGroup;       // 0x02A0 (0x0008)
[0x0000000000000000]
struct FName        LiveReplay_PreScore;        // 0x02A8 (0x0008)
[0x0000000000000000]
struct FName        LiveReplay_Slomo;          // 0x02B0 (0x0008)
[0x0000000000000000]
struct FName        LiveReplay_PostGoal;        // 0x02B8 (0x0008)
[0x0000000000000000]
struct FName        LiveReplay_None;          // 0x02C0 (0x0008)
[0x0000000000000000]
float             HighlightReplayDuration;     // 0x02C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float             TimeBeforeHighlightReplay;    // 0x02CC (0x0004)
[0x0000000000000001] (CPF_Edit)
class UReplay_Soccar_TA*   Replay;              // 0x02D0 (0x0008)
[0x000004000002000] (CPF_Transient)
class AActor*        FocusCar;              // 0x02D8 (0x0008)
[0x0000000000002020] (CPF_Net | CPF_Transient)
float              FocusCarChangeTime;        // 0x02E0 (0x0004)
[0x0000000000002000] (CPF_Transient)
class AActor*        FocusBall;              // 0x02E8 (0x0008)
[0x0000000000002020] (CPF_Net | CPF_Transient)
TArray<struct FBallHitInfo> BallTouches;        // 0x02F0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FName        BallName;               // 0x0300 (0x0008)
[0x0000000000002000] (CPF_Transient)
float              ScoreTime;              // 0x0308 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t            ScoreHitIndex;         // 0x030C (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FName        GoalActorName;         // 0x0310 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UGoal_TA*      ScoredGoal;            // 0x0318 (0x0008)
[0x0000000004082028] (CPF_ExportObject | CPF_Net | CPF_Transient | CPF_Component | CPF_EditInline)
unsigned long       bSlomo : 1;             // 0x0320 (0x0004)
[0x000004000002000] [0x00000001] (CPF_Transient)
unsigned long       bReplicatedSlomo : 1;    // 0x0320 (0x0004)
[0x000004100002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long       bSlomoForDefender : 1;   // 0x0320 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)
unsigned long       bAutoSave : 1;           // 0x0320 (0x0004)
[0x0000000000002000] [0x00000008] (CPF_Transient)
int32_t            FocusHitIndex;         // 0x0324 (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<struct FReplayFocusCar> FocusCars;        // 0x0328 (0x0010)

```

```

[0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t FocusCarIdx; // 0x0338 (0x0004)
[0x0000000000002000] (CPF_Transient)
float ReplayStartTime; // 0x033C (0x0004)
[0x0000000000002000] (CPF_Transient)
float BallSpawnTime; // 0x0340 (0x0004)
[0x0000000000002000] (CPF_Transient)
class AGameEvent_Soccar_TA* SoccarGame; // 0x0348
(0x0008) [0x0000000100002020] (CPF_Net | CPF_Transient)
uint8_t ScoredOnTeam; // 0x0350 (0x0001)
[0x0000000000002000] (CPF_Transient)
struct FReplayScoreData ScoreData; // 0x0358 (0x0020)
[0x0000000100002020] (CPF_Net | CPF_Transient)
int32_t PreviousScores[0x2]; // 0x0378 (0x0008)
[0x0000000000002020] (CPF_Net | CPF_Transient)
int32_t ForceCutToFocusActors; // 0x0380 (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
class UCameraConfig_TA* CameraConfig; // 0x0388 (0x0008)
[0x0000800000000000]
struct FScriptDelegate __EventReplayFinished__Delegate; // 0x0390
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventScoreDataChanged__Delegate; // 0x03A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventAutoSaveChanged__Delegate; // 0x03C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventFocusCarChanged__Delegate; // 0x03D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ReplayDirector_TA");
}

return uClassPointer;
};

void HandleStatTickerMessage(class APRI_TA* PRI, class APRI_TA* Victim, class
UStatEvent_TA* StatEvent);
void HandleGaveStat(class AStatFactory_TA* Factory, class APRI_TA* ToPRI, class
UStatEvent_TA* StatEvent, class ABall_TA* Ball, int32_t BallHitIndex, class APRI_TA* Victim,
class UGoal_TA* Goal);
void HandleStats(class APRI_TA* PRI, class UGoal_TA* Goal, class UStatEvent_TA* StatEvent);
void HandleReplayFinished(class UReplay_TA* InReplay);
bool ShouldSlomo();
void UpdateSlomo();
void UpdateFocusActors();
void SetHighlightFocusActors(class AActor* NewCar, class AActor* NewBall, class UGoal_TA*
NewGoal);
void PlayRandomHighlight();

```

```

int32_t GetNextHighlightFrame();
void PrintDebugInfo(class UDebugDrawer* Drawer);
void SetAutoSave();
void SaveUserKeyframe();
void BuildFocusCars();
void SetSlomo(unsigned long bNewSlomo, unsigned long bForceChange);
void eventTick(float DeltaTime);
void eventDestroyed();
class AActor* GetReplayProxy(struct FName ReplayActorName);
float GetReplayTimeSeconds();
void SetFocusActors(class AActor* NewCar, class AActor* NewBall);
float GetReplayStartTime();
void SetSlomoForDefender(class ABall_TA* Ball, int32_t DefendingTeam);
void OnScoreDataChanged();
void GoalScored(class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIndex);
void HandleScoreUpdated(class ATeam_TA* Team);
void HandleAllTeamsCreated(class AGameEvent_Team_TA* TeamGame);
void RecordPlayers();
void HandleGameStateChanged(class AGameEvent_TA* G);
void OnSoccarGameSet();
void SetGameEvent(class AGameEvent_Soccar_TA* InGameEvent);
void eventReplicatedEvent(struct FName VarName);
void EventFocusCarChanged(class AActor* NewFocusCar);
void EventAutoSaveChanged(class AReplayDirector_TA* Director);
void EventScoreDataChanged(class AReplayDirector_TA* Director);
void EventReplayFinished(class AReplayDirector_TA* Director);
};


```

```

// Class TAGame.CameraSettingsActor_TA
// 0x0070 (0x0268 - 0x02D8)
class ACameraSettingsActor_TA : public AReplicationInfo
{
public:
float UploadCameraDelay; // 0x0268 (0x0004)
[0x0000000000000001] (CPF_Edit)
class APRI_TA* PRI; // 0x0270 (0x0008)
[0x0000004100002020] (CPF_Net | CPF_Transient)
struct FProfileCameraSettings ProfileSettings; // 0x0278 (0x001C)
[0x0000008000002020] (CPF_Net | CPF_Transient)
unsigned long bUsingSecondaryCamera : 1; // 0x0294 (0x0004)
[0x0000008000002020] [0x00000001] (CPF_Net | CPF_Transient)
unsigned long bUsingBehindView : 1; // 0x0294 (0x0004)
[0x0000008000002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long bUsingFreecam : 1; // 0x0294 (0x0004)
[0x0000008000002020] [0x00000004] (CPF_Net | CPF_Transient)
unsigned long bDirtySecondaryCamera : 1; // 0x0294 (0x0004)
[0x0000000000000000] [0x00000008]
unsigned long bDirtyBehindView : 1; // 0x0294 (0x0004)
[0x0000000000000000] [0x00000010]
uint8_t CameraPitch; // 0x0298 (0x0001)
[0x0000008000002020] (CPF_Net | CPF_Transient)
uint8_t CameraYaw; // 0x0299 (0x0001)
[0x0000008000002020] (CPF_Net | CPF_Transient)

```

```

float MaxProximityDistance; // 0x029C (0x0004)
[0x0000008000002000] (CPF_Transient)
float LastRotationChangeTime; // 0x02A0 (0x0004)
[0x0000000000000000]
float LastReplicateRotationTime; // 0x02A4 (0x0004)
[0x0000000000000000]
struct FScriptDelegate __EventCameraChanged__Delegate; // 0x02A8
(0x0018) [0x000000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSecondaryCameraChanged__Delegate; // 0x02C0
(0x0018) [0x000000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraSettingsActor_TA");
}

return uClassPointer;
};

class ACamera_X* GetCamera();
bool AllowTargetSelect();
void HandleMaxProximityDistanceChanged();
void ReplicateChangesToServer();
void ServerSetCameraRotationReliable(uint8_t Pitch, uint8_t Yaw);
void ServerSetCameraRotation(uint8_t Pitch, uint8_t Yaw);
void SetCameraRotation(uint8_t Pitch, uint8_t Yaw);
void ServerSetUsingFreecam(unsigned long bUsing);
void SetUsingFreecam(unsigned long bUsing);
void ServerSetUsingBehindView(unsigned long bUsing);
void SetUsingBehindView(unsigned long bUsing);
void ServerSetUsingSecondaryCamera(unsigned long bUsing);
void SetUsingSecondaryCamera(unsigned long bUsing);
void ServerSetCameraSettings(struct FProfileCameraSettings InSettings);
void SetProfileSettings(struct FProfileCameraSettings InSettings);
void eventDestroyed();
void HandleProfileSettings(class UProfileCameraSave_TA* Settings);
void eventOnOwnerChanged();
void eventReplicatedEvent(struct FName VarName);
void EventSecondaryCameraChanged(class ACameraSettingsActor_TA* Actor);
void EventCameraChanged(class ACameraSettingsActor_TA* Actor);
};

// Class TAGame.AudioParamsSuperSonicComponent_TA
// 0x0004 (0x00A4 - 0x00A8)
class UAudioParamsSuperSonicComponent_TA : public UActorComponent_X
{
public:

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioParamsSuperSonicComponent_TA");
}

return uClassPointer;
};

void __AudioParamsSuperSonicComponent_TA__Construct_0x1(class APlayerController_X* _);
static void SetSuperSonic(unsigned long bSuperSonic);
void HandleDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void HandlePawnChange(class APlayerController_X* PC, class ACar_TA* OldPawn, class
ACar_TA* NewPawn);
void HandleSuperSonicChanged(class ACar_TA* Car);
void UpdateNumLocalPlayers(class UGameViewportClient_TA* GVC, int32_t LocalPlayerCount);
void eventConstruct();
};

// Class TAGame.AudioParamsSuperSonicReplayComponent_TA
// 0x000C (0x00A4 - 0x00B0)
class UAudioParamsSuperSonicReplayComponent_TA : public UActorComponent_X
{
public:
class ACar_TA*          ReplayFocus;           // 0x00A8 (0x0008)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.AudioParamsSuperSonicReplayComponent_TA");
}

return uClassPointer;
};

void __AudioParamsSuperSonicReplayComponent_TA__Attached_0x3(class ACamera_TA* C,
class AActor* T);
void __AudioParamsSuperSonicReplayComponent_TA__Attached_0x2(class ACamera_X* C,
class UCameraState_X* CS);
void __AudioParamsSuperSonicReplayComponent_TA__Attached_0x1(class
APlayerController_X* _);
void HandleSuperSonicChanged(class ACar_TA* Car);
void UpdateReplayCameraLocalView(class ACamera_Replay_TA* Camera);
void eventAttached();
};

```

```

// Class TAGame.Camera_TA
// 0x00E8F0 (0x06C8 - 0x07B08)
class ACamera_TA : public ACamera_X
{
public:
    struct FSwivelExtent           SwivelExtentSlow;          // 0x06C8 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FSwivelExtent           SwivelExtentFast;         // 0x06D4 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    float                          SwivelFastSpeed;        // 0x06E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                          SwivelDieRate;          // 0x06E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FProfileSliderLimits   FOVLimits;                // 0x06E8 (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   HeightLimits;            // 0x06F4 (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   AngleLimits;             // 0x0700 (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   DistanceLimits;          // 0x070C (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   StiffnessLimits;         // 0x0718 (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   SwivelSpeedLimits;        // 0x0724 (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   TransitionSpeedLimits;   // 0x0730 (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    struct FProfileSliderLimits   PrespawnLerpLimits;       // 0x073C (0x000C)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    TArray<struct FProfileCameraSettings> CameraPresetSettings; // 0x07408
    (0x0010) [0x000000000040003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
    float                         HorizontalSplitscreenHeightOffset; // 0x07508 (0x0004)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    float                         HorizontalSplitscreenFOVOffset; // 0x0754C (0x0004)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    float                         VerticalSplitscreenFOVOffset; // 0x075860 (0x0004)
    [0x0000000000000003] (CPF_Edit | CPF_Const)
    float                         ClipRate;                 // 0x075C64 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    struct FRotator               CurrentSwivel;           // 0x07608 (0x000C)
    [0x0000000000002000] (CPF_Transient)
    class ARBActor_TA*           Demolisher;              // 0x07708 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    unsigned long                 bDemolished : 1;        // 0x07780 (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    unsigned long                 bGroundClampTrace : 1; // 0x07780 (0x0004)
    [0x0000000000000002] [0x00000002] (CPF_Const)
    class ACameraSettingsActor_TA* CameraSettings;        // 0x07808
    (0x0008) [0x0000000000002000] (CPF_Transient)
    float                         GroundClampZOffset;      // 0x07890 (0x0004)
    [0x0000000000000002] (CPF_Const)
    class UCameraConfig_TA*       CameraConfig;            // 0x07908 (0x0008)
    [0x0000800000000000]

```

```

struct FScriptDelegate           __EventCameraTargetChanged__Delegate;      // 0x0798A0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Camera_TA");
}

return uClassPointer;
};

void eventOnViewTargetChanged();
float ClipToField(float CameraLocationZ);
void ApplyCameraModifiers(float DeltaTime, struct FPOV& OutPOV);
void Demolished(class ARBActor_TA* InDemolisher);
void ApplyRearCameraView(struct FCameraOrientation& OutPOV);
void ApplySwivel(struct FCameraOrientation& OutPOV);
struct FRotator GetDesiredSwivel(float LookUp, float LookRight);
void UpdateSwivel(float DeltaTime);
float GetDefaultFOVOffset();
float GetDefaultViewHeightOffset();
void UpdateAspectRatio(struct FCameraOrientation& OutPOV);
void PostProcessCameraState(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateFOV();
class ACameraSettingsActor_TA* GetCameraSettings();
void ProcessCameraState(float DeltaTime, struct FCameraOrientation& OutPOV);
void HandleRemoveLocalFX(class USeqAct_AttachFX_TA* SeqAct);
void HandleAddLocalFX(class USeqAct_AttachFX_TA* SeqAct);
void InitializeFor(class APlayerController* PC);
void EventCameraTargetChanged(class ACamera_TA* Camera, class AActor* Target);
};

// Class TAGame.Camera_Replay_TA
// 0x00C0 (0x07B08 - 0x08708)
class ACamera_Replay_TA : public ACamera_TA
{
public:
TArray<class UCameraState_X*>          OverrideStates;           // 0x07B08
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString                         FocusActorString;        // 0x07C08 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FName                           CameraMode;            // 0x07D08 (0x0008)
[0x0008000000000000]
struct FName                           DefaultCameraMode;    // 0x07D8E0 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FProfileSliderLimits           DOFDistanceLimits;   // 0x07E08 (0x000C)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FProfileSliderLimits           DOFIInfLimits;       // 0x07ECF4 (0x000C)
[0x0000000000000003] (CPF_Edit | CPF_Const)

```

```

struct FProfileSliderLimits           DOFKernelLimits;          // 0x07F8800
(0x000C) [0x0000000000000003] (CPF_Edit | CPF_Const)
struct FProfileSliderLimits           GreenScreenLimits;       // 0x0804C (0x000C)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FProfileSliderLimits           BloomScaleLimits;        // 0x08108 (0x000C)
[0x0000000000000003] (CPF_Edit | CPF_Const)
struct FProfileSliderLimits           FilterOverlayLimits;    // 0x081C24 (0x000C)
[0x0000000000000003] (CPF_Edit | CPF_Const)
unsigned long                      bReplayTranslucency : 1;   // 0x082830 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long                      bGreenScreen : 1;         // 0x082830 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
float                             GreenScreenColorIndex;     // 0x082C34 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UCameraState_Replay_TA*      CurrentCameraState;        // 0x08308
(0x0008) [0x0000000000002000] (CPF_Transient)
class UCameraState_X*              PreviousCameraProxyState; // 0x083840
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate             __EventCameraProxyStateChanged__Delegate; // 0x08408 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __CameraMode__ChangeNotify;    // 0x085860
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Camera_Replay_TA");
}

return uClassPointer;
};

bool __Camera_Replay_TA__InstanceCameraStates_0x1(class UCameraState_X* S);
void __CameraMode__ChangeNotifyFunc();
float ClipToField(float CameraLocationZ);
void UpdateAspectRatio(struct FCameraOrientation& OutPOV);
void UpdateFOV();
void ApplySwivel(struct FCameraOrientation& OutPOV);
class AActor* GetFocusActor();
class ACar_TA* GetFocusCar();
class APRI_TA* GetFocusPRI();
class ABall_TA* GetFocusBall();
void SetFocusActor(class FString InFocusActorString);
void SetCameraMode(struct FName Mode);
void UpdateCameraState();
class UCameraState_X* GetCameraStateByName(struct FName ModeName);
void eventUpdateCamera(float DeltaTime);
bool ShouldHaveCameraTrackState();
void InstanceCameraStates();
void EventCameraProxyStateChanged(class UCameraState_Replay_TA* NewCameraState, class

```

```
UCameraState_X* NewProxyState);
};

// Class TAGame.CameraState_TA
// 0x000C (0x008C - 0x0098)
class UCameraState_TA : public UCameraState_X
{
public:
    unsigned long          bShowCarHUD : 1;           // 0x0090 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long          bAllowRearCamera : 1;        // 0x0090 (0x0004)
    [0x0000000000000002] [0x00000002] (CPF_Const)
    float                 StateStartTime;             // 0x0094 (0x0004)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CameraState_TA");
        }

        return uClassPointer;
    }

    void BeginCameraState();
};

// Class TAGame.CameraState_Replay_TA
// 0x001C (0x0098 - 0x00B4)
class UCameraState_Replay_TA : public UCameraState_TA
{
public:
    unsigned long          bLocalMovement : 1;           // 0x0098 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long          bClipToField : 1;            // 0x0098 (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    unsigned long          bUIHighlightFocus : 1;        // 0x0098 (0x0004)
    [0x0000000000000001] [0x00000004] (CPF_Edit)
    class AActor*          OldFocusActor;              // 0x00A0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    struct FVector         OldFocusActorLocation;       // 0x00A8 (0x000C)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
```

```

uClassPointer = UObject::FindClass("Class TAGame.CameraState_Replay_TA");
}

return uClassPointer;
};

class AActor* GetFocusActor();
void UpdateFlyPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateFocusActorPOV(class AActor* FocusActor, float DeltaTime, struct
FCameraOrientation& OutPOV);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void BeginCameraState();
class UCameraState_X* GetProxyCameraState();
bool ShouldClipToField();
bool AllowSwivel();
bool IsDisabled(struct FName CameraMode, class FString InFocusActorString);
};

// Class TAGame.CameraState_ReplayPlayerView_TA
// 0x0024 (0x00B4 - 0x00D8)
class UCameraState_ReplayPlayerView_TA : public UCameraState_Replay_TA
{
public:
class UCameraState_X* [0x0000000000000001] (CPF_Edit) CarCameraState; // 0x00B8 (0x0008)
class UCameraState_X* [0x0000000000000001] (CPF_Edit) BallCameraState; // 0x00C0 (0x0008)
class UCameraState_X* [0x0000000000002000] (CPF_Transient) CurrentCameraState; // 0x00C8 (0x0008)
class UProfileCameraSave_TA* [0x0000000000002000] (CPF_Transient) CameraSave; // 0x00D0 (0x0008)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_ReplayPlayerView_TA");
}

return uClassPointer;
};

void __CameraState_ReplayPlayerView_TA__Init_0x1(class UProfileCameraSave_TA* SO);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
class UCameraState_X* GetProxyCameraState();
void Init(class ACamera_X* InCamera);
bool AllowSwivel();
bool IsDisabled(struct FName CameraMode, class FString InFocusActorString);
};

// Class TAGame.CameraState_DirectorProxy_TA

```

```

// 0x00CC (0x00B4 - 0x0180)
class UCameraState_DirectorProxy_TA : public UCameraState_Replay_TA
{
public:
class AGameEvent_Soccar_TA*           GameEvent;          // 0x00B8
[0x0008] [0x00180000000001] (CPF_Edit)
class UGameObserver_TA*              GameObserver;       // 0x00C0 (0x0008)
[0x0001800004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
float                         MinimumStateTime;    // 0x00C8 (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FInterpCurveFloat            DistToBallWeight;   // 0x00D0 (0x0018)
[0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat            PlayersInViewWeight; // 0x00E8 (0x0018)
[0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat            ViewPitchWeight;   // 0x0100 (0x0018)
[0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
float                          GoalInViewWeight;  // 0x0118 (0x0004)
[0x0001000000000001] (CPF_Edit)
float                          CanSeeNewViewContinuity; // 0x011C (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FInterpCurveFloat            SimilarOrientationContinuity; // 0x0120
(0x0018) [0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat            SimilarCarVelocityContinuity; // 0x0138
(0x0018) [0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FVector                   FieldForward;      // 0x0150 (0x000C)
[0x0001008000002000] (CPF_Transient)
struct FVector                   FieldSide;        // 0x015C (0x000C)
[0x0001008000002000] (CPF_Transient)
unsigned long                    bActive : 1;      // 0x0168 (0x0004)
[0x0001000000000000] [0x00000001]
class UCameraState_Director_TA*     Director;         // 0x0170 (0x0008)
[0x0001000000002000] (CPF_Transient)
class UProfileCameraSave_TA*       CameraSave;       // 0x0178 (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_DirectorProxy_TA");
}

return uClassPointer;
};

void __CameraState_DirectorProxy_TA__Init_0x1(class UProfileCameraSave_TA* SO);
class AActor* GetFocusActor();
float GetCarContinuityRating(struct FVector ViewStart, struct FRotator ViewRotation, float FOV,
struct FVector CarVelocity, struct FVector OtherCarVelocity);
float GetContinuityRating(struct FVector ViewStart, struct FRotator ViewRotation, float FOV);
float GetViewRating(struct FVector ViewStart, struct FRotator ViewRotation, float FOV);

```

```

float GetAspectRatio();
bool AllowSwivel();
bool ShouldKeepExecuting();
float GetActiveTime();
void Init(class ACamera_X* InCamera);
void OnSetGameObserver();
void EndCameraState();
void BeginCameraState();
};

// Class TAGame.CameraState_DirectorPlayerView_TA
// 0x00D4 (0x0180 - 0x0254)
class UCameraState_DirectorPlayerView_TA : public UCameraState_DirectorProxy_TA
{
public:
    class UCameraState_CarRef_TA*           BallCameraState;           // 0x0180
    (0x0008) [0x0001000000000001] (CPF_Edit)
    float                         MinimumHit;                  // 0x0188 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MinimumHitSeparation; // 0x018C (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         StaleDistance;      // 0x0190 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MinimumTimeToBall; // 0x0194 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         CarChangeTimePadding; // 0x0198 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    struct FViewTargetTransitionParams   CarChangeBlendParams; // 0x019C
    (0x0010) [0x0001000000000001] (CPF_Edit)
    struct FInterpCurveFloat          CarChangeBlendTimeForDistance; // 0x01B0
    (0x0018) [0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
    float                         MinBlendContinuity; // 0x01C8 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MaxBlendDistance; // 0x01CC (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MaintainCurrentDistance; // 0x01D0 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MaintainCurrentHitBias; // 0x01D4 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MaintainBallScorability; // 0x01D8 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         MaintainBallScorabilityViewRating; // 0x01DC (0x0004)
    [0x0001000000000001] (CPF_Edit)
    struct FInterpCurveFloat          ContinuityBias;        // 0x01E0 (0x0018)
    [0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
    float                         MinimumViewRating; // 0x01F8 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         AttackerBias;       // 0x01FC (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         AttackingGoalDistance; // 0x0200 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         BadFocusDistance; // 0x0204 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float                         BadFocusViewRating; // 0x0208 (0x0004)
};

```

```

[0x0001000000000001] (CPF_Edit)
float                               BadFocusViewRatingDuration;           // 0x020C (0x0004)
[0x0001000000000001] (CPF_Edit)
class ACar_TA*                      FocusCar;                         // 0x0210 (0x0008)
[0x0001004000002000] (CPF_Transient)
float                               LastCarChangeTime;                // 0x0218 (0x0004)
[0x0001000000002000] (CPF_Transient)
unsigned long                      bIsRelevant : 1;                  // 0x021C (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)
unsigned long                      bBlending : 1;                   // 0x021C (0x0004)
[0x0001000000002000] [0x00000002] (CPF_Transient)
unsigned long                      bExceptionallyBadFocus : 1;      // 0x021C (0x0004)
[0x0001000000002000] [0x00000004] (CPF_Transient)
unsigned long                      bBadViewRating : 1;              // 0x021C (0x0004)
[0x0001000000002000] [0x00000008] (CPF_Transient)
float                             BlendTime;                     // 0x0220 (0x0004)
[0x0001000000002000] (CPF_Transient)
struct FCameraOrientation          SnapshotPOV;                    // 0x0224 (0x002C)
[0x0001000000002000] (CPF_Transient)
float                             BadViewRatingStart;            // 0x0250 (0x0004)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_DirectorPlayerView_TA");
}

return uClassPointer;
};

bool IsCarAttacking(class ACar_TA* Car);
void HandleFocusCarDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldKeepExecuting();
bool ShouldExecute();
float TimeSinceChange();
bool CanConsiderCar(struct FCarData Data);
void FindFocusCar();
void SetFocusCar(class ACar_TA* InCar);
void EndCameraState();
void BeginCameraState();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.PlayerController_Replay_TA
// 0x0010 (0x0980 - 0x0990)
class APlayerController_Replay_TA : public APlayerControllerBase_TA
{
public:

```

```
class AHUD*                               SplitscreenHUDArchetype;           // 0x0980 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AGameEvent_TA*                      GameEvent;                         // 0x0988 (0x0008)
[0x0000800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerController_Replay_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
void eventDestroyed();
void eventReceivedPlayer();
};

// Class TAGame.AudioSlomoReplayComponent_TA
// 0x003C (0x00A4 - 0x00E0)
class UAudioSlomoReplayComponent_TA : public UActorComponent_X
{
public:
struct FName          ReplayPlaybackSpeedStateGroup;    // 0x00A8
(0x0008) [0x0000000000000000]
struct FName          ReplayPlaybackSpeed;           // 0x00B0 (0x0008)
[0x0000000000000000]
struct FName          ReplayPlaybackSpeed01;        // 0x00B8 (0x0008)
[0x0000000000000000]
struct FName          ReplayPlaybackSpeed02;        // 0x00C0 (0x0008)
[0x0000000000000000]
struct FName          ReplayPlaybackSpeed03;        // 0x00C8 (0x0008)
[0x0000000000000000]
struct FName          ReplayPlaybackSpeed04;        // 0x00D0 (0x0008)
[0x0000000000000000]
struct FName          ReplayPlaybackSpeed_05;       // 0x00D8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioSlomoReplayComponent_TA");
}

return uClassPointer;
};
```

```

};

void HandleSlomoChanged();
void eventDetached();
void eventAttached();
};

// Class TAGame.AudioSpectatorMixComponent_TA
// 0x0084 (0x00A4 - 0x0128)
class UAudioSpectatorMixComponent_TA : public UActorComponent_X
{
public:
struct FName [0x0008] [0x0000000000000000] SpectatorCameraModeStateGroup; // 0x00A8
struct FName [0x0000000000000000] SpectatorCameraMode_Fly; // 0x00B0 (0x0008)
struct FName [0x0000000000000000] SpectatorCameraMode_Director_Base; // 0x00B8
struct FName [0x0000000000000000] SpectatorCameraMode_Director_Countdown; // 0x00C0
struct FName [0x0000000000000000] SpectatorCameraMode_None; // 0x00C8
struct FName [0x0000000000000000] CurrentGameState; // 0x00D0 (0x0008)
struct FName [0x0000000000000000] CurrentCameraState; // 0x00D8 (0x0008)
struct FName [0x0000000000000000] ActorCameraFocusRTCP; // 0x00E0 (0x0008)
struct FName [0x0000000000000000] ActorInViewRTPC; // 0x00E8 (0x0008)
struct FName [0x0000000000000000] ReplayCarFocusRTPC; // 0x00F0 (0x0008)
struct FName [0x0000000000000000] ReplayCameraViewFocusRTPC; // 0x00F8
struct FName [0x0000000000000000] IsLocalRTPC; // 0x0100 (0x0008)
class ACar_TA* [0x0000000000000000] TargetCar; // 0x0108 (0x0008)
unsigned long [0x0000000000000000] [0x00000001] bShouldFocusCar : 1; // 0x0110 (0x0004)
unsigned long [0x0000000000000000] [0x00000002] bShouldFocusCameraView : 1; // 0x0110 (0x0004)
unsigned long [0x0000000000000000] [0x00000004] bShouldFocusCarBeLocal : 1; // 0x0110 (0x0004)
class ACamera_Replay_TA* [0x0000000000000000] ReplayCamera; // 0x0118 (0x0008)
class AGameEvent_Soccar_TA* [0x0000000000000000] GameEvent; // 0x0120 (0x0008)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AudioSpectatorMixComponent_TA");
}

return uClassPointer;
};

void __AudioSpectatorMixComponent_TA__ClearIsLocalRTPCs_0x1(class ACar_TA* C);
void __AudioSpectatorMixComponent_TA__ClearInCameraViewRTPCs_0x1(class ACar_TA* C);
void SetAkState(struct FName InState);
void ClearInCameraViewRTPCs();
void ClearIsLocalRTPCs();
void UpdateInCameraViewRTPCs();
void ClearReplayCarFocusRTPCs();
void SetReplayCarFocusRTPC(class ACar_TA* InCar);
void UpdateCarIsLocal(class ACar_TA* InCar);
void HandleCameraProxyChanged(class UCameraState_Replay_TA* NewCameraState, class UCameraState_X* NewCameraProxyState);
void HandleTargetChanged(class ACamera_TA* InCamera, class AActor* NewTarget);
void HandleCameraModeChanged();
void HandleGameStateChanged(class AGameEvent_TA* NewGameState);
void HandleGameEventSet(class AGameEvent_Soccar_TA* InGameEvent);
void eventTick(float DeltaTime);
void eventDetached();
void eventAttached();
};

// Class TAGame.CameraState_Director_TA
// 0x0050 (0x00B4 - 0x0104)
class UCameraState_Director_TA : public UCameraState_Replay_TA
{
public:
class AGameEvent_Soccar_TA* GameEvent; // 0x00B8
(0x0008) [0x0001800000000001] (CPF_Edit)
class UCameraStateSelector_TA* IdleSelector; // 0x00C0 (0x0008)
[0x0001000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCameraStateSelector_TA* CountdownSelector; // 0x00C8
(0x0008) [0x0001000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCameraStateSelector_TA* KickoffSelector; // 0x00D0
(0x0008) [0x0001000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCameraStateSelector_TA* DefaultSelector; // 0x00D8
(0x0008) [0x0001000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FProfileSliderLimits MinTransitionTimeLimits; // 0x00E0
(0x000C) [0x0001000000000003] (CPF_Edit | CPF_Const)
class UCameraStateSelector_TA* CurrentSelector; // 0x00F0
(0x0008) [0x0001000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UCameraState_X* currentState; // 0x00F8 (0x0008)
[0x0001000000000000]
unsigned long bInitialized : 1; // 0x0100 (0x0004)
[0x0001000000000000] [0x00000001]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Director_TA");
}

return uClassPointer;
};

bool IsDisabled(struct FName CameraMode, class FString InFocusActorString);
bool ShouldClipToField();
void UpdateSelector();
void SetSelector(class UCameraStateSelector_TA* InSelector);
class UCameraState_X* GetProxyCameraState();
void InitCameraState(class UCameraState_X* InState);
void InitSelector(class UCameraStateSelector_TA* InSelector);
void InitSelectors();
void InitDirector();
};

// Class TAGame.AutoTourConfig_TA
// 0x0028 (0x0078 - 0x00A0)
class UAutoTourConfig_TA : public UOnlineConfig_X
{
public:
int32_t MaxTournamentsPerWeek; // 0x0078 (0x0004)
[0x0001000000000000]
int32_t TourResultsTimeoutSeconds; // 0x007C (0x0004)
[0x0001000000000000]
int32_t CurrencyID; // 0x0080 (0x0004)
[0x0001000000000000]
int32_t MaxNumPlayers; // 0x0084 (0x0004)
[0x0001000000000000]
unsigned long bAutoPartyUpEnabled : 1; // 0x0088 (0x0004)
[0x0001000000000000] [0x00000001]
TArray<struct FScheduleRegion> ScheduleRegions; // 0x0090
(0x0010) [0x0001000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.AutoTourConfig_TA");
}

return uClassPointer;
};

```

```

};

class FString GetScheduleRegionLabel(class FString RegionID);
};

// Class TAGame.PsyNetService_AutoTourFound_TA
// 0x0050 (0x0090 - 0x00E0)
class UPsyNetService_AutoTourFound_TA : public UPsyNetClientService_X
{
public:
    class UTourSettings_TA*           Tournament;          // 0x0090 (0x0008)
    [0x0001000000000000]
    struct FTourTeam                 Team;              // 0x0098 (0x0048)
    [0x000100000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_AutoTourFound_TA");
        }

        return uClassPointer;
    }
};

// Class TAGame.RPC_TourGetScheduleRegion_TA
// 0x0058 (0x00E8 - 0x0140)
class URPC_TourGetScheduleRegion_TA : public URPC_X
{
public:
    struct FUniqueNetId             PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    class FString                   ScheduleRegion;     // 0x0130 (0x0010)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetScheduleRegion_TA");
        }

        return uClassPointer;
    }
};

class URPC_TourGetScheduleRegion_TA* SetPlayerID(struct FUniqueNetId InPlayerId);

```

```

};

// Class TAGame.RPC_TourSetScheduleRegion_TA
// 0x0058 (0x00E8 - 0x0140)
class URPC_TourSetScheduleRegion_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    class FString                 ScheduleRegion;    // 0x0130 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourSetScheduleRegion_TA");
        }

        return uClassPointer;
    }

    class URPC_TourSetScheduleRegion_TA* SetScheduleRegion(class FString InScheduleRegion);
    class URPC_TourSetScheduleRegion_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_AutoTour_GetSchedule_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_AutoTour_GetSchedule_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    class FString                 Region;            // 0x0130 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)
    TArray<struct FScheduledTournament>   Schedules;        // 0x0140
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_AutoTour_GetSchedule_TA");
        }

        return uClassPointer;
    }
}

```

```

class URPC_AutoTour_GetSchedule_TA* SetRegion(class FString InRegion);
class URPC_AutoTour_GetSchedule_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_AutoTour_GetCycleData_TA
// 0x0098 (0x00E8 - 0x0180)
class URPC_AutoTour_GetCycleData_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    uint64_t                      CycleID;          // 0x0130 (0x0008)
    [0x0001000000002000] (CPF_Transient)
    uint64_t                      CycleEndTime;     // 0x0138 (0x0008)
    [0x0001000000002000] (CPF_Transient)
    int32_t                        WeekID;           // 0x0140 (0x0004)
    [0x0001000000002000] (CPF_Transient)
    uint64_t                        WeekEndTime;      // 0x0148 (0x0008)
    [0x0001000000002000] (CPF_Transient)
    TArray<struct FShopCurrencyInfo> WeeklyCurrencies; // 0x0150
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FTournamentWeek>   Weeks;           // 0x0160 (0x0010)
    [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<int32_t>                 MaxTierScores;    // 0x0170 (0x0010)
    [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_AutoTour_GetCycleData_TA");
        }

        return uClassPointer;
    };

    class URPC_AutoTour_GetCycleData_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.TournamentCompletedEvent_TA
// 0x0014 (0x0060 - 0x0074)
class UTournamentCompletedEvent_TA : public UObject
{
public:
    uint64_t                      TourID;           // 0x0060 (0x0008)
    [0x0001004000000000]
    uint64_t                      ScheduleID;       // 0x0068 (0x0008)
    [0x0001004000000000]
    unsigned long                  bEliminatedFirstRound : 1; // 0x0070 (0x0004)
    [0x0001004000000000] [0x00000001]
}

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TournamentCompletedEvent_TA");
}

return uClassPointer;
};

// Class TAGame.BackFillPolicy_TA
// 0x000C (0x0060 - 0x006C)
class UBackFillPolicy_TA : public UObject
{
public:
uint8_t PolicyType; // 0x0060 (0x0001)
[0x0000000000000001] (CPF_Edit)
int32_t MinSecRemaining; // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t MaxScoreDiff; // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BackFillPolicy_TA");
}

return uClassPointer;
};

bool ShouldBackfill_Knockout(class UOnlineGameDedicatedServerRegistration_X*
DedicatedServer);
bool ShouldBackfill_Soccar(class UOnlineGameDedicatedServerRegistration_X*
DedicatedServer);
void GetBackfillAmount(class UOnlineGameDedicatedServerRegistration_X* DedicatedServer,
int32_t& BackfillTeam1, int32_t& BackfillTeam2);
};

// Class TAGame.BallCamTarget_TA
// 0x0004 (0x00C4 - 0x00C8)
class UBallCamTarget_TA : public UTarget_TA
{
public:

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallCamTarget_TA");
}

return uClassPointer;
};

};

// Class TAGame.GoalExplosionOrientation_TA
// 0x0000 (0x0268 - 0x0268)
class AGoalExplosionOrientation_TA : public AActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GoalExplosionOrientation_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_MapProductAssetOverride_TA
// 0x0018 (0x0060 - 0x0078)
class UProductAttribute_MapProductAssetOverride_TA : public UProductAssetAttribute_TA
{
public:
class UMapSet_TA*           MapsToUseDefaultProduct;          // 0x0060
(0x0008) [0x0000000000000001] (CPF_Edit)
TArray<class UClass*>       GameTypesToUseDefaultProduct;    // 0x0068
(0x0010) [0x00000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
```

```

TAGame.ProductAttribute_MapProductAssetOverride_TA");
}

return uClassPointer;
};

bool ShouldUseDefaultProductasset(class AWorldInfo* CurWorldInfo);
};

// Class TAGame.Ball_GameEditor_TA
// 0x00D8 (0xA48 - 0xB20)
class ABall_GameEditor_TA : public ABall_TA
{
public:
    struct FVector StartLocation; // 0xA48 (0x00C)
    [0x0000000000000000]
    struct FRotator StartRotation; // 0xA54 (0x00C)
    [0x0000000000000000]
    float VelocityStartSpeed; // 0xA60 (0x004)
    [0x0000000000000000]
    struct FRotator VelocityStartRotation; // 0xA64 (0x00C)
    [0x0000000000000000]
    float BallResetTime; // 0xA70 (0x004)
    [0x0000000000000000]
    float FirstHitTime; // 0xA74 (0x004)
    [0x0000000000000000]
    struct FVector CameraPositionOffset; // 0xA78 (0x00C)
    [0x0000000000000000]
    class AFXActor_X* EditingFXActorArchetype; // 0xA88 (0x008)
    [0x0000000000000001] (CPF_Edit)
    class AFXActor_X* EditingFXActor; // 0xA90 (0x008)
    [0x000008000002000] (CPF_Transient)
    unsigned long bUpdateTrajectory : 1; // 0xA98 (0x004)
    [0x0000000000000000] [0x00000001]
    struct FVector ToLocation; // 0xA9C (0x00C)
    [0x0000000000000000]
    struct FScriptDelegate __EventBallWentThruRing__Delegate; // 0xAA8
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __EventVelocityStartSpeedChanged__Delegate; //
    0xAC0 (0x0018) [0x000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __EventFire__Delegate; // 0xAD8 (0x0018)
    [0x000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __EventReset__Delegate; // 0xAF0 (0x0018)
    [0x000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate __EventSelectionChange__Delegate; // 0xB08
    (0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {

```

```
uClassPointer = UObject::FindClass("Class TAGame.Ball_GameEditor_TA");
}

return uClassPointer;
};

void EditingEnd();
void EditingBegin();
void NotifyOnSelectionChange(struct FScriptDelegate Callback);
struct FVector GetCameraPositionOffset();
struct FVector CollisionExtent();
void ModfiyBlendedRotation(struct FRotator ControllerRot, struct FRotator DeltaRot, struct
FRotator& out_Rotation);
bool CanEdit();
struct FRotator EditorGetRotation();
struct FVector EditorGetLocation();
bool SerializedDataIsEqual(class FString S0, class FString S1);
void Deserialize(class UJsonObject* Data);
class FString Serialize();
struct FVector GetGrabRotationOffset(struct FRotator CurrentRotation);
void eventTick(float DeltaTime);
struct FRotator GetDesiredRotation();
void EditorSetRotationFromInput(float Forward, float Strafe);
void EditorSetRotation(struct FRotator NewRotation);
bool EditorSetLocation(struct FVector NewLocation);
bool EditorMoveToLocation(struct FVector Loc);
void WentThruRing();
void OnWentThruRing();
void NotifyIfWentThruRing();
bool IsRoundActive();
void OnCarTouch(class ACar_TA* HitCar, uint8_t HitType);
void OnHitWorld(struct FVector HitLoc, struct FVector HitNormal, class UPhysicalMaterial*
PhysMat);
void eventReset();
void Fire();
struct FVector GetBallFireVelocity();
void AddVelocityStartSpeed(float Amount);
void AddVelocityStartRotation(struct FRotator Amount, float MaxDegrees);
void RecordCarHit(class ACar_TA* HitCar, struct FVector HitLocation, struct FVector HitNormal,
uint8_t HitType);
bool ShouldDrawTrajectory();
struct FVector GetTrajectoryStartVelocity();
struct FRotator GetTrajectoryStartRotation();
struct FVector GetTrajectoryStartLocation();
bool CanEverShowTrajectory();
void InitFX();
void eventPostBeginPlay();
void EventSelectionChange(class UInterface_GameEditor_TA* Object, unsigned long bSelected);
void EventReset(class ABall_GameEditor_TA* Ball);
void EventFire(class ABall_GameEditor_TA* Ball);
void EventVelocityStartSpeedChanged(class ABall_GameEditor_TA* Ball);
void EventBallWentThruRing(class ABall_GameEditor_TA* Ball);
};
```

```

// Class TAGame.Interface_GameEditor_TA
// 0x0000 (0x0060 - 0x0060)
class UInterface_GameEditor_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Interface_GameEditor_TA");
}

return uClassPointer;
};

void EditingEnd();
void EditingBegin();
void NotifyOnSelectionChange(struct FScriptDelegate Callback);
bool SerializedDataIsEqual(class FString S0, class FString S1);
void Deserialize(class UJsonObject* Data);
class FString Serialize();
struct FVector CollisionExtent();
struct FVector GetCameraPositionOffset();
struct FVector GetGrabRotationOffset(struct FRotator CurrentRotation);
void ModifyBlendedRotation(struct FRotator ControllerRot, struct FRotator DeltaRot, struct FRotator& out_Rotation);
struct FRotator GetDesiredRotation();
struct FVector EditorGetLocation();
bool EditorSetLocation(struct FVector NewLocation);
bool EditorMoveToLocation(struct FVector NewLocation);
bool CanEdit();
struct FRotator EditorGetRotation();
void EditorSetRotation(struct FRotator NewRotation);
void EditorSetRotationFromInput(float Forward, float Strafe);
void EventSelectionChange(class UInterface_GameEditor_TA* Object, unsigned long bSelected);
};

// Class TAGame.GameSettingPlaylist_TA
// 0x0070 (0x0160 - 0x01D0)
class UGameSettingPlaylist_TA : public UGameSettingPlaylist_X
{
public:

TArray<struct FMapSkillSettings> MapSkillRequirements; // 0x0160
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UClass* TourServerInfoClass; // 0x0170 (0x0008)
[0x0001000000000001] (CPF_Edit)
unsigned long bAllowHonorDuelChallenges : 1; // 0x0178 (0x0004)
[0x0001000000000001] [0x00000001] (CPF_Edit)
unsigned long bAllowVoiceChat : 1; // 0x0178 (0x0004)
[0x0001000000000001] [0x00000002] (CPF_Edit)

```

```
int32_t DetailsGroup; // 0x017C (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FTeamColor> TeamColors; // 0x0180 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
struct FColorOverride BallSuperSonicColor; // 0x0190 (0x0014)
[0x0000000000000000]
struct FClientLoadoutData LoadoutOverrides[0x2]; // 0x01A8
(0x0020) [0x0000000000040000] (CPF_NeedCtorLink)
class UBackFillPolicy_TA* BackFillPolicy; // 0x01C8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameSettingPlaylist_TA");
}

return uClassPointer;
};

bool HasBackfillPolicy();
void GetBackfillAmount(class UOnlineGameDedicatedServerRegistration_X* DS, int32_t& BackfillTeam1, int32_t& BackfillTeam2);
void Setup(class UPlaylistSettings_X* Settings);
};

// Class TAGame.Ball_Tutorial_TA
// 0x0000 (0xA48 - 0xA48)
class ABall_Tutorial_TA : public ABall_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Ball_Tutorial_TA");
}

return uClassPointer;
};

void eventPostBeginPlay();
};

// Class TAGame.SeqEvent_BallInRange_TA
// 0x0004 (0x017C - 0x0180)
```

```

class USeqEvent_BallInRange_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_BallInRange_TA");
}

return uClassPointer;
};

};

// Class TAGame.BallSlomoComponent_TA
// 0x001C (0x00A4 - 0x00C0)
class UBallSlomoComponent_TA : public UActorComponent_X
{
public:
float BallSloMoRadius; // 0x00A8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float BallSloMoDuration; // 0x00AC (0x0004)
[0x0000000000002000] (CPF_Transient)
float BallSloMoDilation; // 0x00B0 (0x0004)
[0x0000000000002000] (CPF_Transient)
float BallSloMoCooldown; // 0x00B4 (0x0004)
[0x0000000000002000] (CPF_Transient)
float BallSloMoNext; // 0x00B8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float BallSloMoDiffSpeed; // 0x00BC (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BallSlomoComponent_TA");
}

return uClassPointer;
};

void RestoreTimeDilation();
void eventTick(float DeltaTime);
};

```

```
// Class TAGame.SplineLoftActorSpawnable_TA
// 0x0000 (0x0338 - 0x0338)
class ASplineLoftActorSpawnable_TA : public ASplineLoftActorMovable
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SplineLoftActorSpawnable_TA");
}

return uClassPointer;
};

};

// Class TAGame.BanSave_TA
// 0x0020 (0x00C8 - 0x00E8)
class UBanSave_TA : public USaveObject_TA
{
public:
unsigned long           bViewedLastChanceModal : 1;          // 0x00C8 (0x0004)
[0x0008000000000000] [0x00000001]
struct FScriptDelegate   __bViewedLastChanceModal__ChangeNotify; // 
0x00D0 (0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BanSave_TA");
}

return uClassPointer;
};

void __bViewedLastChanceModal__ChangeNotifyFunc();

};

// Class TAGame.RPC_GetBanStatus_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_GetBanStatus_TA : public URPC_X
{
public:
TArray<struct FUniqueNetId>      Players;          // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
```

```

TArray<class UBanMessage_X*>           BanMessages;          // 0x00F8
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetBanStatus_TA");
}

return uClassPointer;
};

class URPC_GetBanStatus_TA* SetPlayers(TArray<struct FUniqueNetId>& InPlayers);
};

// Class TAGame.BeamTraceComponent_TA
// 0x003C (0x00A4 - 0x00E0)
class UBeamTraceComponent_TA : public UActorComponent_X
{
public:
TArray<class UParticleSystemComponent*>   BeamSystems;        // 0x00A8
(0x0010) [0x0000000004480009] (CPF_Edit | CPF_ExportObject | CPF_Component |
CPF_NeedCtorLink | CPF>EditInline)
float                         TraceDistance;      // 0x00B8 (0x0004)
[0x0000000000000001] (CPF>Edit)
float                         MinRetraceRate;    // 0x00BC (0x0004)
[0x0000000000000001] (CPF>Edit)
float                         MaxRetraceRate;    // 0x00C0 (0x0004)
[0x0000000000000001] (CPF>Edit)
TArray<struct FBeamTraceProps>   Beams;            // 0x00C8 (0x0010)
[0x000000000482000] (CPF_Transient | CPF_Component | CPF_NeedCtorLink)
class AFXActor_X*               FXOwner;          // 0x00D8 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BeamTraceComponent_TA");
}

return uClassPointer;
};

void eventTick(float DeltaTime);
void eventDetached();
void eventAttached();

```

```

};

// Class TAGame.BlogTileCache_TA
// 0x0020 (0x00C8 - 0x00E8)
class UBlogTileCache_TA : public USaveObject_TA
{
public:
TArray<class FString> UnreadTileIDs; // 0x00C8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<class FString> PreviouslyDisplayedTileIDs; // 0x00D8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BlogTileCache_TA");
}

return uClassPointer;
};

bool __BlogTileCache_TA__UpdateTileIDs_0x1(class FString ActiveTileID);
void SetAsRead(class FString TileID);
bool IsUnread(class FString TileID);
void UpdateTileIDs(TArray<class FString>& ActiveTileIDs);
};

// Class TAGame.BlogTileConfig_X
// 0x0038 (0x0060 - 0x0098)
class UBlogTileConfig_X : public UObject
{
public:
class UTexture2D* LocalImage; // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
class FString Title; // 0x0068 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString DescriptionHeader; // 0x0078 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString Description; // 0x0088 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BlogTileConfig_X");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.BlueprintErrors_TA
// 0x0010 (0x0080 - 0x0090)
class UBlueprintErrors_TA : public UErrorList
{
public:
class UErrorType* BlueprintNotEnoughCurrency; // 0x0080
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType* BlueprintPriceMismatch; // 0x0088 (0x0008)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BlueprintErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.BlueprintThumbnailLoadResult
// 0x0010 (0x0060 - 0x0070)
class UBlueprintThumbnailLoadResult : public UObject
{
public:
class UTexture* ProductThumbnail; // 0x0060 (0x0008)
[0x0001000000000000]
class UTexture* BlueprintThumbnail; // 0x0068 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BlueprintThumbnailLoadResult");
}

return uClassPointer;
};

```

```
};

// Class TAGame.BotReplacement_TA
// 0x0008 (0x0060 - 0x0068)
class UBotReplacement_TA : public UObject
{
public:
class APRI_TA*           ReplacingBotPRI;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BotReplacement_TA");
}

return uClassPointer;
};

};

// Class TAGame.BreakoutPlatformCollision_TA
// 0x0000 (0x0300 - 0x0300)
class UBreakoutPlatformCollision_TA : public UStaticMeshComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BreakoutPlatformCollision_TA");
}

return uClassPointer;
};

};

// Class TAGame.BreakoutFloor_TA
// 0x0000 (0x0288 - 0x0288)
class ABreakoutFloor_TA : public AStaticMeshActor
{
public:

public:
static UClass* StaticClass()
```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.BreakoutFloor_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.BTAction_BlockGoal  
// 0x0004 (0x01C8 - 0x01CC)  
class UBTAction_BlockGoal : public UBTAction_MoveTo  
{  
public:  
float GoalRadius; // 0x01C8 (0x0004)  
[0x0000000000000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.BTAction_BlockGoal");  
}  
  
return uClassPointer;  
};  
  
uint8_t OnTargetExecute();  
void OnInit();  
};  
  
// Class TAGame.BTAction_DivertBall  
// 0x0000 (0x01E8 - 0x01E8)  
class UBTAction_DivertBall : public UBTAction_HitBall  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.BTAction_DivertBall");  
}
```

```

return uClassPointer;
};

void OnInit();
};

// Class TAGame.BTAction_Jump
// 0x000C (0x0160 - 0x016C)
class UBTAction_Jump : public UBTAction_Target
{
public:
    float MinJumpTime; // 0x0160 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float TargetRadiusHeightScale; // 0x0164 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float StateStartTime; // 0x0168 (0x0004)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTAction_Jump");
        }
    }

    return uClassPointer;
};

float GetImpactTime();
float GetStateTime();
void GotoJumpState(struct FName StateName);
uint8_t eventOnStart();
};

// Class TAGame.BTAction_DodgeHit
// 0x0024 (0x016C - 0x0190)
class UBTAction_DodgeHit : public UBTAction_Jump
{
public:
    float FlipAmount; // 0x0170 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UBTTTarget* AimTarget; // 0x0178 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    struct FVector DodgeDirection; // 0x0180 (0x000C)
    [0x0000000000002000] (CPF_Transient)
    float DodgeTime; // 0x018C (0x0004)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_DodgeHit");
}

return uClassPointer;
};

struct FVector CalculateDodgeDirection();
void SetInputs();
struct FVector GetDodgeVelocity(struct FVector DodgeInput);
uint8_t eventOnStart();
float GetImpactTime();
};

// Class TAGame.BTAction_Face
// 0x001C (0x0160 - 0x017C)
class UBTAction_Face : public UBTAction_Target
{
public:
float ReachAngle; // 0x0160 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxSpeed; // 0x0164 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxTravel; // 0x0168 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bMovingForward : 1; // 0x016C (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
struct FVector StartLocation; // 0x0170 (0x000C)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_Face");
}

return uClassPointer;
};

uint8_t OnTargetExecute();
void SetMovingForward(unsigned long bForward);
uint8_t eventOnStart();
};

// Class TAGame.BTAction_PowerSlide
// 0x0018 (0x0160 - 0x0178)
class UBTAction_PowerSlide : public UBTAction_Target

```

```

{
public:
unsigned long          bReverse : 1;           // 0x0160 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
float                  Delay;                // 0x0164 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UBTTarget*       TurnTarget;           // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                  Steer;                // 0x0170 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t                startYaw;             // 0x0174 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_PowerSlide");
}

return uClassPointer;
};

uint8_t OnTargetExecute();
uint8_t eventOnStart();
};

// Class TAGame.BTAction_Test
// 0x0018 (0x01C8 - 0x01E0)
class UBTAction_Test : public UBTAction_MoveTo
{
public:
struct FVector          TargetLocation;        // 0x01C8 (0x000C)
[0x0000000000002000] (CPF_Transient)
struct FVector          AimLocation;          // 0x01D4 (0x000C)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_Test");
}

return uClassPointer;
};

bool GetAimDir(struct FVector Destination, struct FVector& AimDir);

```

```

uint8_t OnTargetExecute();
uint8_t eventOnStart();
struct FVector GetRandomFieldLocation();
};

// Class TAGame.BTAction_WallBounce
// 0x0004 (0x00BC - 0x00C0)
class UBTAction_WallBounce : public UBTAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTAction_WallBounce");
}

return uClassPointer;
};

};

// Class TAGame.BTC_AIPlayerIndex
// 0x0010 (0x008C - 0x009C)
class UBTC_AIPlayerIndex : public UBTCondition
{
public:
int32_t IndexGreaterThan; // 0x0090 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t IndexLessThan; // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t IndexEqualTo; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_AIPlayerIndex");
}

return uClassPointer;
};

bool eventGetResult();
void HandleMembersChanged(class ATeam_TA* Team, class APRI_TA* PRI);
void eventOnInit();

```

```
};

// Class TAGame.BTC_AreaClear
// 0x0010 (0x01D0 - 0x01E0)
class UBTC_AreaClear : public UBTC_TargetBase
{
public:
    class UBTTarget*           Source;          // 0x01D0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float                      SourcePredictionTime; // 0x01D8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                      Radius;          // 0x01DC (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTC_AreaClear");
        }

        return uClassPointer;
    }

    bool eventGetResult();
    void eventOnInit();
};

// Class TAGame.BTC_BallWelded
// 0x0004 (0x008C - 0x0090)
class UBTC_BallWelded : public UBTCondition
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTC_BallWelded");
        }

        return uClassPointer;
    }

    bool eventGetResult();
};

// Class TAGame.BTC_Between
```

```
// 0x0018 (0x00E8 - 0x0100)
class UBTC_Between : public UBTC_ProxyBase
{
public:
class UBTTarget* Start; // 0x00E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UBTTarget* End; // 0x00F0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float AngleLessThan; // 0x00F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AngleGreaterThan; // 0x00FC (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Between");
}

return uClassPointer;
};

bool eventGetResult();

// Class TAGame.BTC_Boost
// 0x000C (0x00E8 - 0x00F4)
class UBTC_Boost : public UBTC_ProxyBase
{
public:
float BoostAmountLessThan; // 0x00E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostAmountGreaterThan; // 0x00EC (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bFull : 1; // 0x00F0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bNotFull : 1; // 0x00F0 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Boost");
}

return uClassPointer;
}
```

```
};

bool eventGetResult();
};

// Class TAGame.BTC_BounceHeight
// 0x0008 (0x01D0 - 0x01D8)
class UBTC_BounceHeight : public UBTC_TargetBase
{
public:
float HeightLessThan; // 0x01D0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float HeightGreaterThan; // 0x01D4 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_BounceHeight");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_CanJumpTo
// 0x0000 (0x01D8 - 0x01D8)
class UBTC_CanJumpTo : public UBTC_BounceHeight
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_CanJumpTo");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_ClosestTime
```

```
// 0x0008 (0x0218 - 0x0220)
class UBTC_ClosestTime : public UBTC_Target
{
public:
float TimeGreaterThan; // 0x0218 (0x0004)
[0x0000000000000001] (CPF_Edit)
float TimeLessThan; // 0x021C (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_ClosestTime");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_ClosestTo
// 0x000C (0x01D0 - 0x01DC)
class UBTC_ClosestTo : public UBTC_TargetBase
{
public:
class UBTTTarget* Targets; // 0x01D0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float ExtraDistance; // 0x01D8 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_ClosestTo");
}

return uClassPointer;
};

bool eventGetResult();
void eventOnInit();
};

// Class TAGame.BTC_JumpHit
// 0x0004 (0x01D0 - 0x01D4)
```

```
class UBTC_JumpHit : public UBTC_TargetBase
{
public:
float MinImpactTime; // 0x01D0 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_JumpHit");
}

return uClassPointer;
};

bool eventGetResult();
void PredictJump(float Time);
};

// Class TAGame.BTC_DodgeHit
// 0x000C (0x01D4 - 0x01E0)
class UBTC_DodgeHit : public UBTC_JumpHit
{
public:
float FlipAmount; // 0x01D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float JumpTime; // 0x01DC (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_DodgeHit");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_Facing
// 0x0000 (0x0218 - 0x0218)
class UBTC_Facing : public UBTC_Target
{
public:
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Facing");
}

return uClassPointer;
};

};

// Class TAGame.BTC_FieldDistance
// 0x0108 (0x01D0 - 0x02D8)
class UBTC_FieldDistance : public UBTC_TargetBase
{
public:
class UBTTTarget* Goal; // 0x01D0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UBTTTarget* OtherGoal; // 0x01D8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float ForwardGreaterThanOrEqual; // 0x01E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ForwardLessThan; // 0x01E4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideGreaterThanOrEqual; // 0x01E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SideLessThan; // 0x01EC (0x0004)
[0x0000000000000001] (CPF_Edit)
float AbsSideGreaterThanOrEqual; // 0x01F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AbsSideLessThan; // 0x01F4 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bUseGoalOrientation : 1; // 0x01F8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
struct FAIProxyData GoalData; // 0x0200 (0x0048)
[0x0000000000002000] (CPF_Transient)
struct FAIProxyData OtherGoalData; // 0x0248 (0x0048)
[0x0000000000002000] (CPF_Transient)
struct FAIProxyData MidFieldData; // 0x0290 (0x0048)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_FieldDistance");
}

```

```

}

return uClassPointer;
};

bool eventGetResult();
struct FVector GetRelativeOffset(struct FVector InLocation);
void eventOnInit();
};

// Class TAGame.BTC_GameInfoType
// 0x001C (0x008C - 0x00A8)
class UBTC_GameInfoType : public UBTCondition
{
public:
TArray<class UClass*>           ValidGameInfoClasses;          // 0x0090 (0x0010)
[0x0000000000040000] (CPF_Edit | CPF_NeedCtorLink)
class AGameInfo_TA*              CurrentGameInfo;            // 0x00A0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_GameInfoType");
}

return uClassPointer;
};

bool eventGetResult();
void eventOnInit();
};

// Class TAGame.BTC_GameScore
// 0x0010 (0x008C - 0x009C)
class UBTC_GameScore : public UBTCondition
{
public:
float                         ScoreDeltaGreaterThan;        // 0x0090 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                         ScoreDeltaLessThan;         // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t                        ScoreDelta;                // 0x0098 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}
}
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_GameScore");
}

return uClassPointer;
};

bool eventGetResult();
void HandleGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class
UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
void eventOnInit();
};

// Class TAGame.BTC_GameState
// 0x000C (0x008C - 0x0098)
class UBTC_GameState : public UBTCondition
{
public:
struct FName StateName; // 0x0090 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_GameState");
}

return uClassPointer;
};

bool eventGetResult();
void HandleGameStateChanged(class AGameEvent_TA* GameEvent);
void eventOnInit();
};

// Class TAGame.BTC_HitLanding
// 0x0000 (0x01D0 - 0x01D0)
class UBTC_HitLanding : public UBTC_TargetBase
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_HitLanding");
}

```

```
}

return uClassPointer;
};

bool eventGetResult();
void OnInit();
};

// Class TAGame.BTC_HumanPlayerOnOpposingTeam
// 0x0004 (0x008C - 0x0090)
class UBTC_HumanPlayerOnOpposingTeam : public UBTCondition
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_HumanPlayerOnOpposingTeam");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_HumanPlayerOnTeam
// 0x0004 (0x008C - 0x0090)
class UBTC_HumanPlayerOnTeam : public UBTCondition
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_HumanPlayerOnTeam");
}

return uClassPointer;
};

bool eventGetResult();
void HandleMembersChanged(class ATeam_TA* Team, class APRI_TA* PRI);
void eventOnInit();
};
```

```
// Class TAGame.BTC_Impact
// 0x0008 (0x0218 - 0x0220)
class UBTC_Impact : public UBTC_Target
{
public:
float TimeGreaterThanOrEqual; // 0x0218 (0x0004)
[0x0000000000000001] (CPF_Edit)
float TimeLessThanOrEqual; // 0x021C (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Impact");
}

return uClassPointer;
};

bool eventGetResult();
};

// Class TAGame.BTC_InCone
// 0x0018 (0x00E8 - 0x0100)
class UBTC_InCone : public UBTC_ProxyBase
{
public:
class UBTTTarget* Source; // 0x00E8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UBTTTarget* Destination; // 0x00F0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float AngleLessThanOrEqual; // 0x00F8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AngleGreaterThanOrEqual; // 0x00FC (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_InCone");
}

return uClassPointer;
};
```

```
bool eventGetResult();
};

// Class TAGame.BTC_Landing
// 0x0008 (0x0218 - 0x0220)
class UBTC_Landing : public UBTCTarget
{
public:
    float LandTimeLessThan; // 0x0218 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float LandTimeGreaterThan; // 0x021C (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTC_Landing");
        }

        return uClassPointer;
    }

    bool eventGetResult();
};

// Class TAGame.BTC_Lock
// 0x0010 (0x008C - 0x009C)
class UBTC_Lock : public UBTCondition
{
public:
    struct FName LockName; // 0x0090 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    int32_t NumLocks; // 0x0098 (0x0004)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTC_Lock");
        }

        return uClassPointer;
    }

    bool eventGetResult();
    void HandleLockRemoved(struct FName InLockName);
}
```

```

void HandleLockAdded(struct FName InLockName);
void ListenForLocks(class UBTLockCollection* Locks);
void eventOnInit();
};

// Class TAGame.BTC_SegmentDistance
// 0x0028 (0x00E8 - 0x0110)
class UBTC_SegmentDistance : public UBTC_ProxyBase
{
public:
    class UBTTTarget* Start; // 0x00E8 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UBTTTarget* End; // 0x00F0 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float StartGreaterThanOrEqual; // 0x00F8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float StartLessThan; // 0x00FC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float EndGreaterThanOrEqual; // 0x0100 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float EndLessThan; // 0x0104 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float AbsSideGreaterThanOrEqual; // 0x0108 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float AbsSideLessThan; // 0x010C (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTC_SegmentDistance");
        }

        return uClassPointer;
    }

    bool eventGetResult();
};

// Class TAGame.BTC_Skill
// 0x000C (0x008C - 0x0098)
class UBTC_Skill : public UBTCondition
{
public:
    float SkillGreaterThanOrEqual; // 0x0090 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float SkillLessThan; // 0x0094 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Skill");
}

return uClassPointer;
};

bool eventGetResult();
void HandleSkillChanged(class AAIController_TA* InAI);
void eventOnInit();
};

// Class TAGame.BTC_SoccarState
// 0x0010 (0x0098 - 0x00A8)
class UBTC_SoccarState : public UBTC_GameState
{
public:
unsigned long          bKickoff : 1;           // 0x0098 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class AGameEvent_Soccar_TA*      SoccarGame;        // 0x00A0
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_SoccarState");
}

return uClassPointer;
};

bool eventGetResult();
void HandleBallHit(class AGameEvent_Soccar_TA* G);
void eventOnInit();
};

// Class TAGame.BTC_Stuck
// 0x0004 (0x008C - 0x0090)
class UBTC_Stuck : public UBTCondition
{
public:

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Stuck");
}

return uClassPointer;
};

bool IsTouchingOtherCar();
bool eventGetResult();
};

// Class TAGame.BTC_TargetNumberSatisfying
// 0x0014 (0x008C - 0x00A0)
class UBTC_TargetNumberSatisfying : public UBTCondition
{
public:
class UBTTarget* Targets; // 0x0090 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t TargetsGreaterThanOrEqual; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t TargetsLessThan; // 0x009C (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_TargetNumberSatisfying");
}

return uClassPointer;
};

bool eventGetResult();
void eventOnInit();
};

// Class TAGame.BTC_TeamSize
// 0x000C (0x008C - 0x0098)
class UBTC_TeamSize : public UBTCondition
{
public:
int32_t MinTeamSize; // 0x0090 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t MaxTeamSize; // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_TeamSize");
}

return uClassPointer;
};

bool eventGetResult();
void HandleMembersChanged(class ATeam_TA* Team, class APRI_TA* PRI);
void eventOnInit();
};

// Class TAGame.BTC_TimeSinceBallHit
// 0x0020 (0x008C - 0x00AC)
class UBTC_TimeSinceBallHit : public UBTCondition
{
public:
uint8_t CheckTeam; // 0x0090 (0x0001)
[0x0000000000000001] (CPF_Edit)
float TimeGreaterThan; // 0x0094 (0x0004)
[0x0000000000000001] (CPF_Edit)
float TimeLessThan; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
class ABall_TA* Ball; // 0x00A0 (0x0008)
[0x0000000000002000] (CPF_Transient)
float LastHitTime; // 0x00A8 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_TimeSinceBallHit");
}

return uClassPointer;
};

bool eventGetResult();
void HandleCarTouch(class ABall_TA* InBall, class ACar_TA* HitCar, uint8_t HitType);
void HandleBallChanged(class AAIController_Soccar_TA* InAI);
void OnInit();
};

// Class TAGame.BTC_Traits
// 0x0004 (0x008C - 0x0090)

```

```

class UBTC_Traits : public UBTCondition
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Traits");
}

return uClassPointer;
};

};

// Class TAGame.BTC_Vehicle
// 0x0014 (0x00E8 - 0x00FC)
class UBTC_Vehicle : public UBTC_ProxyBase
{
public:
unsigned long          bOnBack : 1;                                // 0x00E8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bNotOnBack : 1;                             // 0x00E8 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long          bWheelsOnGround : 1;                         // 0x00E8 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long          bWheelsNotOnGround : 1;                      // 0x00E8 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
float                 GroundNormalGreaterThan;                   // 0x00EC (0x0004)
[0x0000000000000001] (CPF_Edit)
float                 GroundNormalLessThan;                     // 0x00F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                 ForwardSpeedGreaterThan;                  // 0x00F4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                 ForwardSpeedLessThan;                    // 0x00F8 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.BTC_Vehicle");
}

return uClassPointer;
};

```

```
bool eventGetResult();
};

// Class TAGame.BTMoveModifier_Boost
// 0x0020 (0x0074 - 0x0094)
class UBTMoveModifier_Boost : public UBTMoveModifier
{
public:
    float MinDistanceFromDestination; // 0x0078 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float MaxAngleFromCar; // 0x007C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float MaxAngleFromDestination; // 0x0080 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class AVehiclePickup_Boost_TA* BoostPickup; // 0x0088 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    float LastCheckTime; // 0x0090 (0x0004)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BTMoveModifier_Boost");
        }

        return uClassPointer;
    }

    class AVehiclePickup_Boost_TA* FindBoostPickup(struct FVector Destination);
    void HandleBoostPickedUp(class AVehiclePickup_TA* Pickup);
    void SetBoostPickup(class AVehiclePickup_Boost_TA* NewPickup);
    void AdjustDestination(struct FVector Goal, struct FVector& Destination);
    };

// Class TAGame.BugReportSync_TA
// 0x0000 (0x0060 - 0x0060)
class UBugReportSync_TA : public UObject
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.BugReportSync_TA");
        }

        return uClassPointer;
    }
}
```

```

return uClassPointer;
};

void HandleBugReportRequest(class UPsyNetService_BugReport_TA* Message);
void eventConstruct();
};

// Class TAGame.PsyNetService_BugReport_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_BugReport_TA : public UPsyNetClientService_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_BugReport_TA");
}

return uClassPointer;
};

};

// Class TAGame.CameraConfig_TA
// 0x0004 (0x0078 - 0x007C)
class UCameraConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bFramerateIndependentInterp : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bReplayFocusOnCarAfterGoal : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)

unsigned long          bEnableDemoSwivel : 1;           // 0x0078 (0x0004)
[0x0001000000000001] [0x00000004] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraConfig_TA");
}

return uClassPointer;
};

```

```
};

// Class TAGame.SeqAct_AttachFX_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_AttachFX_TA : public USequenceAction
{
public:
    class UParticleSystemComponent* PSC; // 0x0160 (0x0008)
    [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_AttachFX_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
};

// Class TAGame.Camera_Edit_TA
// 0x0010 (0x07B08 - 0x07C08)
class ACamera_Edit_TA : public ACamera_TA
{
public:
    struct FName CameraMode; // 0x07B08 (0x0008)
    [0x0000000000000000]

    class UCameraState_Edit_TA* CurrentCameraState; // 0x07B8C0
    (0x0008) [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.Camera_Edit_TA");
        }

        return uClassPointer;
    }

    void UpdateCameraState();
    void eventUpdateCamera(float DeltaTime);
    void InstanceCameraStates();
};
```

```
// Class TAGame.CameraState_Edit_TA
// 0x0004 (0x0098 - 0x009C)
class UCameraState_Edit_TA : public UCameraState_TA
{
public:
unsigned long           bLocalMovement : 1;           // 0x0098 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Edit_TA");
}

return uClassPointer;
};

void UpdateFlyPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void BeginCameraState();
class UCameraState_X* GetProxyCameraState();
bool AllowSwivel();
};

// Class TAGame.Camera_GameEditor_TA
// 0x0010 (0x07B08 - 0x07C08)
class ACamera_GameEditor_TA : public ACamera_TA
{
public:
struct FName           CameraMode;           // 0x07B08 (0x0008)
[0x0000000000000000]
class UCameraState_GameEditor_TA*      CurrentCameraState;      //
0x07B8C0 (0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Camera_GameEditor_TA");
}

return uClassPointer;
};

void UpdateCameraState();
void eventUpdateCamera(float DeltaTime);
```

```

void InstanceCameraStates();
};

// Class TAGame.CameraState_GameEditor_TA
// 0x0000 (0x0098 - 0x0098)
class UCameraState_GameEditor_TA : public UCameraState_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_GameEditor_TA");
}

return uClassPointer;
};

void UpdateFlyPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void BeginCameraState();
class UCameraState_X* GetProxyCameraState();
bool AllowSwivel();
};

// Class TAGame.Camera_MainMenu_TA
// 0x0020 (0x06C8 - 0x06E8)
class ACamera_MainMenu_TA : public ACamera_X
{
public:
struct FVector HandheldCamFrequency; // 0x06C8 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector HandheldCamAmplitude; // 0x06D4 (0x000C)
[0x0000000000000001] (CPF_Edit)
class UCameraState_X* CurrentState; // 0x06E0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Camera_MainMenu_TA");
}

return uClassPointer;
};

```

```

float GetAspectRatio();
void PostProcessCameraState(float DeltaTime, struct FCameraOrientation& OutPOV);
void ApplyCameraModifiers(float DeltaTime, struct FTPOV& OutPOV);
class UCameraState_X* SelectCameraState();
void SetCameraState(struct FName StateName);
class UCameraState_X* GetCameraStateByName(struct FName StateName);
void InstanceCameraStates();
};

// Class TAGame.CameraState_CarPreview_TA
// 0x0084 (0x008C - 0x0110)
class UCameraState_CarPreview_TA : public UCameraState_X
{
public:
TArray<float> SideOffsets; // 0x0090 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FCameraOrientation> ViewOffsets; // 0x00A0
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float BlendRate; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinPitch; // 0x00B4 (0x0004)
[0x0000000000000000]
float MaxPitch; // 0x00B8 (0x0004)
[0x0000000000000000]
struct FCameraOrientation CurrentOrientation; // 0x00BC (0x002C)
[0x0000000000002000] (CPF_Transient)
float CurrentSideOffset; // 0x00E8 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FRotator RotationOffset; // 0x00EC (0x000C)
[0x0000000000002000] (CPF_Transient)
class UCameraState_PremiumGarage_TA* PrevPremiumGarageState; // 0x00F8 (0x0008) [0x0000000000002000] (CPF_Transient)
TArray<struct FTargetCacheEntry> TargetsCache; // 0x0100
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_CarPreview_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void AdjustOutPOV CameraRotation(struct FRotator AddedRotator, struct FCameraOrientation& OutPOV);
void OnSnap();
void PreviewFadeIn();
void PreviewFadeOut(class UCameraState_CarPreview_TA* InPrevState);

```

```

void BeginCameraState();
void GetAccumulatedOrientation(struct FVector& out_Location, struct FRotator& out_Rotation,
int32_t& NumPreviewActors);
void GetActorOrientation(struct FCameraOrientation& OutPOV, float& OutSideOffset);
void GetSingleActorOrientation(class ACarPreviewActor_TA* ViewTarget, struct FVector&
OutLoc, struct FRotator& OutRot);
void UpdateCachedTarget(class ACarPreviewActor_TA* Target, struct FVector Location, struct
FRotator Rotation);
bool IsTargetCached(class ACarPreviewActor_TA* Target);
int32_t FindTargetCacheIndex(class ACarPreviewActor_TA* Target);
};

// Class TAGame.ProfileCameraSave_TA
// 0x0088 (0x00C8 - 0x0150)
class UProfileCameraSave_TA : public USaveObject_TA
{
public:
struct FProfileAutoCamSettings           AutoCam;                      // 0x00C8 (0x0028)
[0x0000000000000000]
struct FProfileCameraSettings           Camera;                     // 0x00F0 (0x001C)
[0x0000000000000000]
unsigned long                           bInvertSwivelPitch : 1;      // 0x010C (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long                           bInvertSpectatorPitch : 1;    // 0x010C (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long                           bLockSpectatorBallCam : 1;   // 0x010C (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long                           bFreecam : 1;                // 0x010C (0x0004)
[0x0000000000000000] [0x00000008]
unsigned long                           bPrefersSecondaryCamera : 1; // 0x010C (0x0004)
[0x0000000000000000] [0x00000010]
unsigned long                           bToggleSecondaryCamera : 1;  // 0x010C (0x0004)
[0x0008000000000000] [0x00000020]
unsigned long                           bToggleRearCamera : 1;       // 0x010C (0x0004)
[0x0000000000000000] [0x00000040]
unsigned long                           bUseBallIndicator : 1;       // 0x010C (0x0004)
[0x0000000000000000] [0x00000080]
unsigned long                           bUseBallCamIndicator : 1;   // 0x010C (0x0004)
[0x0000000000000000] [0x00000100]
unsigned long                           bEnableCameraShake : 1;     // 0x010C (0x0004)
[0x0000000000000000] [0x00000200]
unsigned long                           bDirectorAllowCameraBlending : 1; // 0x010C
(0x0004) [0x0000000000000000] [0x00000400]
uint8_t                               CameraSettingsPreset;        // 0x0110 (0x0001)
[0x0000000000004000] (CPF_Config)
float                                 DirectorMinChangeTeamFocusTime; // 0x0114 (0x0004)
[0x0000000000000000]
float                                 DirectorMinSameTeamFocusTime; // 0x0118 (0x0004)
[0x0000000000000000]
float                                 MaxProximityDistance;        // 0x011C (0x0004)
[0x0008000000000000]
struct FScriptDelegate                 _bToggleSecondaryCamera__ChangeNotify; // 0x0120 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 _MaxProximityDistance__ChangeNotify; // 0x0138

```

(0x0018) [0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProfileCameraSave_TA");  
}  
  
return uClassPointer;  
};  
  
void __MaxProximityDistance__ChangeNotifyFunc();  
void __bToggleSecondaryCamera__ChangeNotifyFunc();  
void OnLoad();  
};  
  
// Class TAGame.CameraState_CarRef_TA  
// 0x0010 (0x0098 - 0x00A8)  
class UCameraState_CarRef_TA : public UCameraState_TA  
{  
public:  
class ACar_TA* Car; // 0x0098 (0x0008)  
[0x00000000000000002000] (CPF_Transient)  
class ACameraSettingsActor_TA* CameraSettings; // 0x00A0  
(0x0008) [0x00000000000000002000] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.CameraState_CarRef_TA");  
}  
  
return uClassPointer;  
};  
  
struct FProfileCameraSettings GetProfileCameraSettings();  
struct FRotator GetCarRotation();  
struct FVector GetCarLocation();  
bool IsOnGround();  
bool ShouldExecute();  
float TimeSinceSpawn();  
bool UpdateCarRef();  
};  
  
// Class TAGame.CameraState_Car_TA  
// 0x012C (0x00A8 - 0x01D4)
```

```

class UCameraState_Car_TA : public UCameraState_CarRef_TA
{
public:
    float InterpToAirRate; // 0x00A8 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float InterpToGroundRate; // 0x00AC (0x0004)
    [0x0000000000000001] (CPF_Editable)
    struct FInterpVector FocusInterp; // 0x00B0 (0x0028)
    [0x0000000000000001] (CPF_Editable)
    struct FInterpVector FocusOffsetInterp; // 0x00D8 (0x0028)
    [0x0000000000000001] (CPF_Editable)
    struct FInterpVector DistanceInterp; // 0x0100 (0x0028)
    [0x0000000000000001] (CPF_Editable)
    float GroundRotationInterpRate; // 0x0128 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float GroundRotationInterpRateWall; // 0x012C (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float AirVelocityInfluence; // 0x0130 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float AirVelocityInfluenceMaxSpeed; // 0x0134 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float DistanceSpeedScale; // 0x0138 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float DistanceOffsetMin; // 0x013C (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float MaxSpeedFOV; // 0x0140 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float FOVInterpSpeed; // 0x0144 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float SupersonicFOV; // 0x0148 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float SupersonicFOVInterpSpeed; // 0x014C (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float BouncyChassisScale; // 0x0150 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float RollScale; // 0x0154 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float GroundNormalInterpRate; // 0x0158 (0x0004)
    [0x0000000000000001] (CPF_Editable)
    float AirGroundBlend; // 0x015C (0x0004)
    [0x0000000000002000] (CPF_Transient)
    struct FCameraOrientation ViewOffset; // 0x0160 (0x002C)
    [0x0000000000002000] (CPF_Transient)
    unsigned long bFirstExecution : 1; // 0x018C (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    unsigned long bOnGround : 1; // 0x018C (0x0004)
    [0x0000000000002000] [0x00000002] (CPF_Transient)
    struct FVector GroundNormal; // 0x0190 (0x000C)
    [0x0000000000002000] (CPF_Transient)
    float CameraStartDistance; // 0x019C (0x0004)
    [0x0000000000000000]
    class UCameraConfig_TA* Config; // 0x01A0 (0x0008)
    [0x0000800000000000]
    float ProximityDistance; // 0x01A8 (0x0004)

```

```

[0x0000000000000000] (CPF_Transient)
float           ProximityDetectionSpeedCap;          // 0x01AC (0x0004)
[0x0000000000000002] (CPF_Const)
float           ProximityDistanceMultiplier;        // 0x01B0 (0x0004)
[0x0000000000000002] (CPF_Const)
float           ProximityDetectionMultiplier;        // 0x01B4 (0x0004)
[0x0000000000000002] (CPF_Const)
float           ProximityDistanceBlendInSpeed;      // 0x01B8 (0x0004)
[0x0000000000000002] (CPF_Const)
float           ProximityDistanceBlendOutSpeed;     // 0x01BC (0x0004)
[0x0000000000000002] (CPF_Const)
class UClass*    ProximityTargetClass;             // 0x01C0 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FProfileSliderLimits   ProximityDistanceLimits; // 0x01C8 (0x000C)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Car_TA");
}

return uClassPointer;
};

static void StaticOverrideBlendParams(float InTransitionSpeedSetting, struct
FViewTargetTransitionParams& OutParams);
void OverrideBlendParams(struct FViewTargetTransitionParams& Params);
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
void UpdateInterpVector(struct FVector NewValue, float DeltaTime, struct FInterpVector& Interp);
void UpdateRotationModifiers(float DeltaTime, struct FCameraOrientation& OutPOV);
void GetNewFOVParameters(float& NewFOV, float& NewInterpSpeed);
void UpdateFOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateDistance(float DeltaTime, struct FCameraOrientation& OutPOV);
void ScalePitch(float SpeedScale, struct FRotator& OutRotation);
struct FRotator CalculateDesiredAirRotation(struct FCameraOrientation& OutPOV);
void UpdateAirPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateGroundPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateAirAndGroundCamera(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateFocus(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateFocusWorldOffset(float DeltaTime);
void UpdateAirGroundBlend(float DeltaTime);
void UpdateViewOffset();
bool ShouldAllowProximityTarget(class UTarget_TA* InTarget);
void UpdateProximityDistance(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateValidPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void ResetView();
void BeginCameraState();
};

```

```

// Class TAGame.CameraState_BallCam_TA
// 0x0094 (0x01D4 - 0x0268)
class UCameraState_BallCam_TA : public UCameraState_Car_TA
{
public:
    float           RotationRate;          // 0x01D8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           PitchScale;            // 0x01DC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           PitchFocusZFactor;     // 0x01E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           PitchExtentMin;       // 0x01E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float           PitchExtentMax;       // 0x01E8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UTTarget_TA*      Target;        // 0x01F0 (0x0008)
    [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    struct FCameraOrientation   BasePOV;      // 0x01F8 (0x002C)
    [0x0000000000002000] (CPF_Transient)
    struct FVector           OldTargetLocation; // 0x0224 (0x000C)
    [0x0000000000002000] (CPF_Transient)
    unsigned long             bWasReverseCam : 1; // 0x0230 (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    unsigned long             bActive : 1;      // 0x0230 (0x0004)
    [0x0000000000000000] [0x00000002]
    TArray<class UClass*>    TargetClasses;    // 0x0238 (0x0010)
    [0x000000000400000] (CPF_NeedCtorLink)
    TArray<struct FPlaneSelectData>  PlaneSelectSettings; // 0x0248
    (0x0010) [0x000000000400000] (CPF_NeedCtorLink)
    TArray<struct FAngleSelectData> AngleSelectSettings; // 0x0258
    (0x0010) [0x000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CameraState_BallCam_TA");
        }

        return uClassPointer;
    };

    bool AllowTargetSelect();
    void UpdateFocus(float DeltaTime, struct FCameraOrientation& OutPOV);
    void UpdateFocusWorldOffset(float DeltaTime);
    void UpdateAirAndGroundCamera(float DeltaTime, struct FCameraOrientation& OutPOV);
    void UpdateRotationModifiers(float DeltaTime, struct FCameraOrientation& OutPOV);
    void UpdateValidPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
    bool ShouldExecute();
    bool TargetIsWelded();
}

```

```

bool ShouldAllowBallCamTarget(class UTarget_TA* InTarget);
class UTarget_TA* GetHUDTarget();
class UTarget_TA* FindBestTarget(class UTarget_TA* OldTarget, TArray<class UClass*>&
InTargetClasses);
bool ChooseTarget();
void SetTarget(class UTarget_TA* NewTarget);
void HandleRemoveTarget(class UObject* NewTarget);
void HandleAddTarget(class UTarget_TA* NewTarget);
void TargetSelect(class APlayerController_TA* PC, int32_t Direction);
class UTarget_TA* NextPotentialTarget(class APlayerController_TA* PC, int32_t Direction);
void EndCameraState();
void BeginCameraState();
void ShutDown();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraSettingsActorCopy_TA
// 0x0000 (0x02D8 - 0x02D8)
class ACameraSettingsActorCopy_TA : public ACameraSettingsActor_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraSettingsActorCopy_TA");
}

return uClassPointer;
};

void eventTick(float DeltaTime);
};

// Class TAGame.TargetFinder_TA
// 0x0018 (0x0060 - 0x0078)
class UTargetFinder_TA : public UObject
{
public:
struct FScriptDelegate           __TargetFilter__Delegate;           // 0x0060 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TargetFinder_TA");
}

```

```

}

return uClassPointer;
};

static bool TargetIsValid(class UTarget_TA* InTarget);
static bool CanUseTarget(class AActor* Owner, class UTarget_TA* InTarget);
static bool RotatedPlaneContainsPoint(struct FVector Origin, struct FRotator PlaneRotation, float YExtent, float ZExtent, struct FVector Point, float& Out_DistanceToOrigin);
static TArray<class UTarget_TA*> FindTargetsInRotatedPlane(class UClass* TargetClass, class AActor* Owner, struct FVector Location, struct FRotator Rotation, class UTarget_TA* ActiveTarget, struct FVector CheckDirection, float YExtent, float ZExtent, float MaxRadius, float MaxDegreeAngle, struct FScriptDelegate Callback_AllowTarget);
static class UTarget_TA* FindClosestTargetInRotatedPlane(class UClass* TargetClass, class AActor* Owner, struct FVector Location, struct FRotator Rotation, class UTarget_TA* ActiveTarget, struct FVector CheckDirection, float YExtent, float ZExtent, float MaxRadius, float MaxDegreeAngle, struct FScriptDelegate Callback_AllowTarget);
static class UTarget_TA* FindClosestTargetInRotatedPlanes(class UClass* TargetClass, class AActor* Owner, struct FVector Location, struct FRotator Rotation, class UTarget_TA* ActiveTarget, struct FVector CheckDirection, struct FScriptDelegate Callback_AllowTarget, TArray<struct FPlaneSelectData>& PlaneSelectSettings);
static class UTarget_TA* FindClosestTargetInDirection(class UClass* TargetClass, class AActor* Owner, struct FVector Location, class UTarget_TA* ActiveTarget, struct FVector ViewDirection, int32_t YDirection, float MaxRadius, struct FScriptDelegate Callback_AllowTarget);
static class UTarget_TA* FindClosestTargetByAngle(class UClass* TargetClass, class AActor* Owner, struct FVector AngleLocation, class UTarget_TA* ActiveTarget, struct FVector CheckDirection, float MaxDegreeAngle, float MaxRadius, struct FScriptDelegate Callback_AllowTarget);
static TArray<class UTarget_TA*> FindTargetsByAngle(class UClass* TargetClass, class AActor* Owner, struct FVector AngleLocation, class UTarget_TA* ActiveTarget, struct FVector CheckDirection, float MaxDegreeAngle, float MaxRadius, struct FScriptDelegate Callback_AllowTarget);
static TArray<class UTarget_TA*> FindTargetsWithinRadius(class UClass* TargetClass, class AActor* Owner, struct FVector Location, class UTarget_TA* ActiveTarget, float MaxRadius, struct FScriptDelegate Callback_AllowTarget);
static class UTarget_TA* FindClosestTargetByAngles(class UClass* TargetClass, class AActor* Owner, struct FVector AngleLocation, class UTarget_TA* ActiveTarget, struct FVector CheckDirection, struct FScriptDelegate Callback_AllowTarget, TArray<struct FAngleSelectData>& AngleSelectSettings);
static class UTarget_TA* FindBestTarget(class UClass* TargetClass, class AActor* Owner, struct FVector Location, struct FRotator Rotation, class UTarget_TA* ActiveTarget, struct FScriptDelegate Callback_AllowTarget, TArray<struct FPlaneSelectData>& PlaneSelectSettings, TArray<struct FAngleSelectData>& AngleSelectSettings);
bool TargetFilter(class UTarget_TA* Target);
};

// Class TAGame.CameraState_BallCam_KnockOut_TA
// 0x0000 (0x0268 - 0x0268)
class UCameraState_BallCam_KnockOut_TA : public UCameraState_BallCam_TA
{
public:

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_BallCam_KnockOut_TA");
}

return uClassPointer;
};

void HandleDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void HandleTargetTeamChange(class APlayerReplicationInfo* PRI);
void SetTarget(class UTarget_TA* NewTarget);
bool ShouldAllowBallCamTarget(class UTarget_TA* InTarget);
};

// Class TAGame.Target_Knockout_TA
// 0x0004 (0x00C4 - 0x00C8)
class UTarget_Knockout_TA : public UTarget_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Target_Knockout_TA");
}

return uClassPointer;
};

struct FVector GetFocus();
};

// Class TAGame.CameraState_BallCamInverted_TA
// 0x0018 (0x0268 - 0x0280)
class UCameraState_BallCamInverted_TA : public UCameraState_BallCam_TA
{
public:
struct FViewTargetTransitionParams           StartBlendParams;           // 0x0268
(0x0010) [0x0000000000000000]
unsigned long          bAllowBlendTimeOverride : 1;           // 0x0278 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
float                DelayFromSpawn;           // 0x027C (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_BallCamInverted_TA");
}

return uClassPointer;
};

void UpdateFocusWorldOffset(float DeltaTime);
struct FViewTargetTransitionParams GetStartBlendParams(class UCameraState_X*
PreviousState);
void OverrideBlendParams(struct FViewTargetTransitionParams& Params);
void UpdateValidPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldExecute();
void EndCameraState();
void BeginCameraState();
};

// Class TAGame.CameraState_CarInverted_TA
// 0x001C (0x01D4 - 0x01F0)
class UCameraState_CarInverted_TA : public UCameraState_Car_TA
{
public:
struct FViewTargetTransitionParams           StartBlendParams;          // 0x01D8
(0x0010) [0x0000000000000000]
unsigned long                  bAllowBlendTimeOverride : 1;      // 0x01E8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
float                         DelayFromSpawn;                // 0x01EC (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_CarInverted_TA");
}

return uClassPointer;
};

void UpdateGroundPOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateFocusWorldOffset(float DeltaTime);
void UpdateViewOffset();
struct FViewTargetTransitionParams GetStartBlendParams(class UCameraState_X*
PreviousState);
void OverrideBlendParams(struct FViewTargetTransitionParams& Params);
bool ShouldExecute();
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
};

```

```

// Class TAGame.CameraState_LiveReplay_TA
// 0x00A0 (0x008C - 0x012C)
class UCameraState_LiveReplay_TA : public UCameraState_X
{
public:
    struct FVector LocationOffset; // 0x0090 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    struct FInterpCurveFloat FocusSpeedOverTime; // 0x00A0
    (0x0018) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    float CarBallFocusBlendDistance; // 0x00B8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float Distance; // 0x00BC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float FOV; // 0x00C0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float SlomoInterpTime; // 0x00C4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float SlomoDistance; // 0x00C8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float SlomoFOV; // 0x00CC (0x0004)
    struct FVector VelocityPrediction; // 0x00D0 (0x000C)
    [0x0000000000000001] (CPF_Edit)
    float InterpToFieldCenterRate; // 0x00DC (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float InterpToFieldCenterOffset; // 0x00E0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float ExplosionFocusDelay; // 0x00E4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float ScorerOffsetInterpSpeed; // 0x00E8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class AGameEvent_Soccar_TA* SoccarGame; // 0x00F0
    (0x0008) [0x0000000000002000] (CPF_Transient)
    class ACar_TA* FocusCar; // 0x00F8 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class ABall_TA* FocusBall; // 0x0100 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    unsigned long bHasDoneFirstUpdate : 1; // 0x0108 (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    unsigned long bCanUpdate : 1; // 0x0108 (0x0004)
    [0x0000000000002000] [0x00000002] (CPF_Transient)
    unsigned long bSetGoalScorerFocusOffset : 1; // 0x0108 (0x0004)
    [0x0000000000002000] [0x00000004] (CPF_Transient)
    float LastFocusChangeTime; // 0x010C (0x0004)
    [0x0000000000002000] (CPF_Transient)
    float SlomoTime; // 0x0110 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    class UGoal_TA* Goal; // 0x0118 (0x0008)
    [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    struct FVector GoalScorerFocusOffset; // 0x0120 (0x000C)
    [0x0000000000002000] (CPF_Transient)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_LiveReplay_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
struct FVector GetCameraBaseLocation(struct FVector CameraFocus);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void Tick(float DeltaTime);
void UpdateSlomo(float DeltaTime);
void EndCameraState();
void BeginCameraState();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
bool ShouldExecute();
};

// Class TAGame.CameraState_BasketballLiveReplay_TA
// 0x0004 (0x012C - 0x0130)
class UCameraState_BasketballLiveReplay_TA : public UCameraState_LiveReplay_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_BasketballLiveReplay_TA");
}

return uClassPointer;
};

struct FVector GetCameraBaseLocation(struct FVector CameraFocus);
};

// Class TAGame.CameraState_BreakoutLiveReplay_TA
// 0x0008 (0x012C - 0x0134)
class UCameraState_BreakoutLiveReplay_TA : public UCameraState_LiveReplay_TA
{
public:
float GoalVerticalOffset; // 0x0130 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_BreakoutLiveReplay_TA");
}

return uClassPointer;
};

struct FVector GetCameraBaseLocation(struct FVector CameraFocus);
};

// Class TAGame.CameraState_CameraTrack_TA
// 0x0034 (0x00B4 - 0x00E8)
class UCameraState_CameraTrack_TA : public UCameraState_Replay_TA
{
public:
class UReplay_TA*           Replay;           // 0x00B8 (0x0008)
[0x0000800000000000]
uint8_t                     FlyType;         // 0x00C0 (0x0001)
[0x0000000000000001] (CPF_Edit)
unsigned long                bFlying : 1;    // 0x00C4 (0x0004)
[0x0000000000000000] [0x00000001]
float                        LastReplayTime; // 0x00C8 (0x0004)
[0x0000000000000000]
struct FRotator              AccumulatedLook; // 0x00CC (0x000C)
[0x0000000000000000]
struct FVector               AccumulatedMove; // 0x00D8 (0x000C)
[0x0000000000000000]
float                        LastFOV;        // 0x00E4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_CameraTrack_TA");
}

return uClassPointer;
};

void SetFlying(unsigned long bFly);
bool AllowFlyingWhenPaused();
bool AllowFlyingWhenMoving();
bool AllowFlying();
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void BeginCameraState();

```

```

};

// Class TAGame.CameraTrack_TA
// 0x0058 (0x0060 - 0x00B8)
class UCameraTrack_TA : public UObject
{
public:
TArray<class UCameraTrackPoint_TA*> Points; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FInterpCurveVector LocationCurve; // 0x0070 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FInterpCurveVector RotationCurve; // 0x0088 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FInterpCurveFloat FOVCurve; // 0x00A0 (0x0018)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraTrack_TA");
}

return uClassPointer;
};

struct FInterpCurvePointFloat __CameraTrack_TA__RebuildCurves_0x3(class
UCameraTrackPoint_TA* P);
struct FInterpCurvePointVector __CameraTrack_TA__RebuildCurves_0x2(struct
FCameraTrackRotationPoint R);
struct FInterpCurvePointVector __CameraTrack_TA__RebuildCurves_0x1(class
UCameraTrackPoint_TA* P);
void ResetCurves();
TArray<struct FCameraTrackRotationPoint> GenerateRotationPoints();
void RebuildCurves();
float GetEndTime();
void GetPOV(float Time, struct FCameraOrientation& OutPOV);
void RemovePOV(int32_t frame);
void InsertPoint(class UCameraTrackPoint_TA* Point);
void RecordPOV(int32_t frame, float Time, struct FCameraOrientation& POV);
};

// Class TAGame.CameraState_Car_KnockOut_TA
// 0x0008 (0x0268 - 0x0270)
class UCameraState_Car_KnockOut_TA : public UCameraState_BallCam_TA
{
public:
class UTarget_TA* PotentialTarget; // 0x0268 (0x0008)
[0x0001004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:

```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Car_KnockOut_TA");
}

return uClassPointer;
};

bool AllowTargetSelect();
struct FRotator CalculateDesiredAirRotation(struct FCameraOrientation& OutPOV);
bool TouchingCameraVolume();
void UpdateAirAndGroundCamera(float DeltaTime, struct FCameraOrientation& OutPOV);
class UTarget_TA* GetHUDTarget();
bool ShouldAllowPotentialTarget(class UTarget_TA* InTarget);
class UTarget_TA* FindPotentialTarget();
void Tick(float DeltaTime);
bool ShouldExecute();
};

// Class TAGame.CameraVolume_KnockOut_TA
// 0x0004 (0x02A4 - 0x02A8)
class ACameraVolume_KnockOut_TA : public AVolume
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraVolume_KnockOut_TA");
}

return uClassPointer;
};

};

// Class TAGame.Target_World_TA
// 0x0004 (0x00C4 - 0x00C8)
class UTarget_World_TA : public UTarget_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Target_World_TA");
}

return uClassPointer;
};

};

// Class TAGame.CameraState_PremiumGarage_TA
// 0x0078 (0x0110 - 0x0188)
class UCameraState_PremiumGarage_TA : public UCameraState_CarPreview_TA
{
public:
struct FViewTargetTransitionParams           BetweenPremiumCameraBlend;          // 0x0110 (0x0010) [0x0000000000000000]
float                                         FadeInFromBlackTime;                // 0x0120 (0x0004)
[0x0000000000000000]                         float                                         FadeToBlackTime;                  // 0x0124 (0x0004)
[0x0000000000000000]                         float                                         PitchDeadzone;                   // 0x0128 (0x0004)
[0x0000000000000000]                         float                                         PitchSensitivity;                // 0x012C (0x0004)
[0x0000000000000000]                         struct FInterpCurveFloat               PitchAutocorrectCurve;            // 0x0130 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)      float                                         PitchMaxSpeed;                  // 0x0148 (0x0004)
[0x0000000000000000]                         float                                         PitchBlendbackModifier;           // 0x014C (0x0004)
[0x0000000000000000]                         float                                         PitchBlendbackMin;                // 0x0150 (0x0004)
[0x0000000000000000]                         struct FInterpCurveFloat               CameraReturnBlendCurve;           // 0x0158
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
float                                         LastAdjustTime;                  // 0x0170 (0x0004)
[0x0000000000002000] (CPF_Transient)        class ATurnTableActor_TA*             TurntableActor;                 // 0x0178 (0x0008)
[0x0000000000002000] (CPF_Transient)        class UGFxShell_TA*                 Shell;                          // 0x0180 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_PremiumGarage_TA");
}

return uClassPointer;
}

```

```

};

void UpdateAllViewTargetRotation(float DeltaTime);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void AdjustOutPOV CameraRotation(struct FRotator AddedRotator, struct FCameraOrientation& OutPOV);
float GetCameraCurveSpeed();
void ManuallyAdjustCar(unsigned long bEnable);
void BeginCameraState();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
void GetActorOrientation(struct FCameraOrientation& OutPOV, float& OutSideOffset);
void GetAccumulatedOrientation(struct FVector& out_Location, struct FRotator& out_Rotation,
int32_t& NumPreviewActors);
};

// Class TAGame.CarPreviewSet_TA
// 0x0068 (0x0060 - 0x00C8)
class UCarPreviewSet_TA : public UObject
{
public:
TArray<class ACarPreviewActor_TA*> CarPreviewActors; // 0x0060
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UOnlineGameParty_TA* OnlineParty; // 0x0070 (0x0008)
[0x0000004000002000] (CPF_Transient)
unsigned long bRemotePlayersHidden : 1; // 0x0078 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
int32_t MaxCars; // 0x007C (0x0004)
[0x0000000000000002] (CPF_Const)
struct FScriptDelegate __EventPreviewActorsChanged__Delegate; // 0x0080
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPreviewActorSetProfile__Delegate; // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSwapFinished__Delegate; // 0x00B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarPreviewSet_TA");
}

return uClassPointer;
};

void __CarPreviewSet_TA__Cleanup_0x1(class ACarPreviewActor_TA* PreviewActor);
bool SetRemotePlayersHidden(unsigned long bHidden);
void HandlePrimaryPlayerSet(class APlayerController_Menu_TA* PC);
void OnPreviewActorsChanged();
void HandlePartyMemberLoadoutRemoved();
void HandlePartyMemberLoadoutChange(struct FUniqueNetId PlayerID);

```

```

void HandleReceiveController(class UPlayer* PlayerRef);
void HandlePlayerJoin(class ULocalPlayer* P);
void UpdatePreview(int32_t Index);
void UpdatePreviewForPlayer(class ULocalPlayer* P);
class ACarPreviewActor_TA* GetPlayerCarPreviewActor(class ULocalPlayer* P);
void UpdateAllPreviews();
void InitPreviewActors();
void HandleCarPreviewActorAdded(class UGameShare_TA* GameShare, class
ACarPreviewActor_TA* PreviewActor);
void Cleanup();
void Init();
void EventSwapFinished();
void EventPreviewActorSetProfile(class ACarPreviewActor_TA* Car);
void EventPreviewActorsChanged(class UCarPreviewSet_TA* G);
};

// Class TAGame.CameraState_Countdown_TA
// 0x0038 (0x0180 - 0x01B8)
class UCameraState_Countdown_TA : public UCameraState_DirectorProxy_TA
{
public:
int32_t MaxCarsShown; // 0x0180 (0x0004)
[0x0001000000000001] (CPF_Edit)
float TimePerCar; // 0x0184 (0x0004)
[0x0001000000000001] (CPF_Edit)
unsigned long bSwitchTeamsEveryCar : 1; // 0x0188 (0x0004)
[0x0001000000000001] [0x00000001] (CPF_Edit)
TArray<class ACar_TA*> ShownCars; // 0x0190 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class ACar_TA* CurrentCar; // 0x01A0 (0x0008)
[0x000100000002000] (CPF_Transient)
class ACar_TA* NextCar; // 0x01A8 (0x0008)
[0x000100000002000] (CPF_Transient)
int32_t LastTeam; // 0x01B0 (0x0004)
[0x000100000002000] (CPF_Transient)
float LastChangeCarTime; // 0x01B4 (0x0004)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Countdown_TA");
}

return uClassPointer;
};

bool __CameraState_Countdown_TA__FindNextCar_0x1(struct FCarData D);
bool __CameraState_Countdown_TA__FindNextCar_0x2(class ACar_TA* C);
void UpdateTextureStreaming();

```

```

void UpdateNextCar();
class ACar_TA* FindNextCar();
bool ShouldKeepExecuting();
bool ShouldExecute();
void EndCameraState();
void BeginCameraState();
};

// Class TAGame.CameraState_CustomFocus_TA
// 0x0024 (0x01D4 - 0x01F8)
class UCameraState_CustomFocus_TA : public UCameraState_Car_TA
{
public:
float RotationRate; // 0x01D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float PitchScale; // 0x01DC (0x0004)
[0x0000000000000001] (CPF_Edit)
float PitchFocusZFactor; // 0x01E0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float PitchExtentMin; // 0x01E4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float PitchExtentMax; // 0x01E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
class ACustomFocusTarget_TA* FocusTarget; // 0x01F0
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_CustomFocus_TA");
}

return uClassPointer;
};

void UpdateFocus(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateFocusWorldOffset(float DeltaTime);
void UpdateAirAndGroundCamera(float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdateRotationModifiers(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldExecute();
void Init(class ACamera_X* InCam);
};

// Class TAGame.CustomFocusTarget_TA
// 0x0000 (0x0268 - 0x0268)
class ACustomFocusTarget_TA : public AActor
{
public:

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CustomFocusTarget_TA");
}

return uClassPointer;
};

struct FVector GetFocusLocation(class ACar_TA* ForCar);
};

// Class TAGame.CameraState_Demolished_TA
// 0x002C (0x0098 - 0x00C4)
class UCameraState_Demolished_TA : public UCameraState_TA
{
public:
float RotationInterpRate; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
class APawn* DemolishedPawn; // 0x00A0 (0x0008)
[0x0000000000002000] (CPF_Transient)
float StateTime; // 0x00A8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float InitialFOV; // 0x00AC (0x0004)
[0x0000000000000001] (CPF_Edit)
float FOVBlendTime; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bInterp : 1; // 0x00B4 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long bTrackDemolisher : 1; // 0x00B4 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
float MaxZoomInFOV; // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxFOVBlendDistance; // 0x00BC (0x0004)
[0x0000000000000001] (CPF_Edit)
float Distance; // 0x00C0 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Demolished_TA");
}

return uClassPointer;
};

```

```
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void EndCameraState();
void BeginCameraState();
bool ShouldKeepExecuting();
bool ShouldExecute();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
};

// Class TAGame.CameraStateSelector_TA
// 0x0018 (0x0070 - 0x0088)
class UCameraStateSelector_TA : public UComponent
{
public:
TArray<class UCameraState_X*> States; // 0x0070 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class ACamera_X* Camera; // 0x0080 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraStateSelector_TA");
}

return uClassPointer;
};

class UCameraState_X* SelectState(class UCameraState_X* InCurrentState);
void SetActive();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraState_DirectorBallCam_TA
// 0x001C (0x0268 - 0x0284)
class UCameraState_DirectorBallCam_TA : public UCameraState_BallCam_TA
{
public:
struct FProfileCameraSettings ForcedCameraSettings; // 0x0268
(0x001C) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_DirectorBallCam_TA");
}
}
```

```

return uClassPointer;
};

struct FProfileCameraSettings GetProfileCameraSettings();
};

// Class TAGame.CameraState_DirectorCountdownRotate_TA
// 0x0020 (0x01B8 - 0x01D8)
class UCameraState_DirectorCountdownRotate_TA : public UCameraState_Countdown_TA
{
public:
    float FOV; // 0x01B8 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    float Distance; // 0x01BC (0x0004)
    [0x0001000000000001] (CPF_Edit)
    struct FRotator InitialRotation; // 0x01C0 (0x000C)
    [0x0001000000000001] (CPF_Edit)
    struct FRotator RotationSpeed; // 0x01CC (0x000C)
    [0x0001000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.CameraState_DirectorCountdownRotate_TA");
        }
    }

    return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
};

// Class TAGame.CameraState_DirectorCountdownTailPass_TA
// 0x0020 (0x01B8 - 0x01D8)
class UCameraState_DirectorCountdownTailPass_TA : public UCameraState_Countdown_TA
{
public:
    float DistanceFromCar; // 0x01B8 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    struct FRotator InitialRotation; // 0x01BC (0x000C)
    [0x0001000000000001] (CPF_Edit)
    struct FRotator RotationSpeed; // 0x01C8 (0x000C)
    [0x0001000000000001] (CPF_Edit)
    float FOV; // 0x01D4 (0x0004)
    [0x0001000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.CameraState_DirectorCountdownTailPass_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
};

// Class TAGame.CameraState_DirectorGoalScorer_TA
// 0x0010 (0x0180 - 0x0190)
class UCameraState_DirectorGoalScorer_TA : public UCameraState_DirectorProxy_TA
{
public:
float Distance; // 0x0180 (0x0004)
[0x0001000000000001] (CPF_Edit)
float FOV; // 0x0184 (0x0004)
[0x0001000000000001] (CPF_Edit)
class ACar_TA* FocusCar; // 0x0188 (0x0008)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_DirectorGoalScorer_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldKeepExecuting();
bool ShouldExecute();
void HandlePlayerScored(class AGameEvent_Soccar_TA* InGameEvent, class APRI_TA* Scorer);
void EndCameraState();
void BeginCameraState();
void ShutDown();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraState_DirectorKickoff_TA
// 0x0024 (0x0180 - 0x01A4)
class UCameraState_DirectorKickoff_TA : public UCameraState_DirectorProxy_TA
{
public:

```

```

float Distance; // 0x0180 (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FRotator RotationOffset; // 0x0184 (0x000C)
[0x0001000000000001] (CPF_Edit)
float FOV; // 0x0190 (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FViewTargetTransitionParams BlendToParams; // 0x0194
(0x0010) [0x0001000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_DirectorKickoff_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
bool ShouldKeepExecuting();
};

// Class TAGame.CameraState_DirectorMoving_TA
// 0x0064 (0x0180 - 0x01E4)
class UCameraState_DirectorMoving_TA : public UCameraState_DirectorProxy_TA
{
public:
struct FVector StartOffset; // 0x0180 (0x000C)
[0x0001000000000001] (CPF_Edit)
struct FVector EndOffset; // 0x018C (0x000C)
[0x0001000000000001] (CPF_Edit)
struct FRotator StartRotation; // 0x0198 (0x000C)
[0x0001000000000001] (CPF_Edit)
struct FRotator EndRotation; // 0x01A4 (0x000C)
[0x0001000000000001] (CPF_Edit)
struct FViewTargetTransitionParams PanBlend; // 0x01B0 (0x0010)
[0x0001000000000001] (CPF_Edit)
float StartFOV; // 0x01C0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float EndFOV; // 0x01C4 (0x0004)
[0x0001000000000001] (CPF_Edit)
uint8_t LoopType; // 0x01C8 (0x0001)
[0x0001000000000001] (CPF_Edit)
struct FVector CachedStart; // 0x01CC (0x000C)
[0x000100000002000] (CPF_Transient)
struct FVector CachedEnd; // 0x01D8 (0x000C)
[0x000100000002000] (CPF_Transient)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_DirectorMoving_TA");
}

return uClassPointer;
};

float GetRawPercent(float inTime);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraState_Freecam_TA
// 0x0084 (0x00A8 - 0x012C)
class UCameraState_Freecam_TA : public UCameraState_CarRef_TA
{
public:
struct FCameraOrientation           ViewOffset_Near;          // 0x00A8 (0x002C)
[0x0000000000000001] (CPF_Edit)
struct FCameraOrientation           ViewOffset_Far;           // 0x00D4 (0x002C)
[0x0000000000000001] (CPF_Edit)
struct FCameraOrientation           ViewOffset;              // 0x0100 (0x002C)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Freecam_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldExecute();
};

// Class TAGame.CameraState_IntroMenu_TA
// 0x004C (0x008C - 0x00D8)
class UCameraState_IntroMenu_TA : public UCameraState_X
{
public:
float                           FOV;                      // 0x0090 (0x0004)
[0x0000000000000000]
struct FVector                  Offset;                 // 0x0094 (0x000C)

```

```

[0x0000000000000000]
float BlendRate; // 0x00A0 (0x0004)
[0x0000000000000000]
class ASkeletalMeshActorMAT_IntroCar_TA* Focus; // 0x00A8
(0x0008) [0x0000004000002000] (CPF_Transient)
TArray<class ASkeletalMeshActorMAT_IntroCar_TA*> FocusActors; //
0x00B0 (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate __EventFocusChanged__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_IntroMenu_TA");
}

return uClassPointer;
};

int32_t FocusActorSort(class ASkeletalMeshActorMAT_IntroCar_TA* A, class
ASkeletalMeshActorMAT_IntroCar_TA* B);
class ASkeletalMeshActorMAT_IntroCar_TA* GetInitialCarFocus();
void SetupFocusActors();
void SetFocus(class ASkeletalMeshActorMAT_IntroCar_TA* NewFocus);
void ShiftFocus(int32_t Direction);
void ShiftLeftFocus();
void ShiftRightFocus();
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void BeginCameraState();
void EventFocusChanged(class UCameraState_IntroMenu_TA* CameraState);
};

// Class TAGame.SkeletalMeshActorMAT_IntroCar_TA
// 0x0018 (0x03E8 - 0x0400)
class ASkeletalMeshActorMAT_IntroCar_TA : public ASkeletalMeshActorMAT_Products_TA
{
public:
int32_t ViewIndex; // 0x03E8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FRotator AngleOffset; // 0x03EC (0x000C)
[0x0000000000000001] (CPF_Edit)
class UIintroCarTemplates_TA* IntroCarsTemplates; // 0x03F8
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.SkeletalMeshActorMAT_IntroCar_TA");
}

return uClassPointer;
};

bool __SkeletalMeshActorMAT_IntroCar_TA__PostBeginPlay_0x1(struct FEditorIntroCar
IntroCar);
void eventPostBeginPlay();
};

// Class TAGame.CameraState_Lobby_TA
// 0x004C (0x008C - 0x00D8)
class UCameraState_Lobby_TA : public UCameraState_X
{
public:
unsigned long           bInitDefaultPOV : 1;           // 0x0090 (0x0004)
[0x0001000000000000] [0x00000001]
TArray<struct FCameraSettings>      TeamSettings;        // 0x0098 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
float                  FadeInDuration;          // 0x00A8 (0x0004)
[0x0001000000000000]
struct FCameraOrientation      DefaultCamera;        // 0x00AC (0x002C)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Lobby_TA");
}

return uClassPointer;
};

bool InitDefaultPOV();
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldKeepExecuting();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
bool ShouldExecute();
};

// Class TAGame.CameraState_LobbyFocus_TA
// 0x0050 (0x008C - 0x00DC)
class UCameraState_LobbyFocus_TA : public UCameraState_X
{
public:
float                  XOffsetDistance;          // 0x0090 (0x0004)
[0x0001000000000000]
float                  CameraOffsetPitch;        // 0x0094 (0x0004)

```

```

[0x0001000000000000]
TArray<float> AngleOffsets; // 0x0098 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class ACar_TA* TargetCar; // 0x00A8 (0x0008)
[0x000100000002000] (CPF_Transient)
struct FCameraOrientation TargetCamera; // 0x00B0 (0x002C)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_LobbyFocus_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void SetTarget(class ACar_TA* Target);
void HandleSelectedFocusChanged(class UPreMatchLobby_TA* PreMatchLobby);
void SetupTargetCamera();
bool ShouldKeepExecuting();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
void BeginCameraState();
bool ShouldExecute();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.PreMatchLobby_TA
// 0x0038 (0x0060 - 0x0098)
class UPreMatchLobby_TA : public UObject
{
public:
class ACar_TA* SelectedFocus; // 0x0060 (0x0008)
[0x0001000000000000]
class ACar_TA* HighlightedFocus; // 0x0068 (0x0008)
[0x0001000000000000]
struct FVector LastHighlightLocation; // 0x0070 (0x000C)
[0x0001000000000000]
struct FScriptDelegate __EventSelectedFocusChanged__Delegate; // 0x0080
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PreMatchLobby_TA");
}

```

```

}

return uClassPointer;
};

class ACar_TA* GetSelectedFocus();
class ACar_TA* GetHighlightedFocus();
class ACar_TA* GetOwnerCar();
class ULocalPlayer_TA* GetLocalPlayer();
class ACar_TA* GetClosestCarToLocation(struct FVector Location);
void SelectHighlightFocusHorizontal(int32_t Direction);
void SelectHighlightFocusVertical(int32_t Direction);
void ToggleSelectedFocusFromHighlight();
void SetHighlighted(class ACar_TA* Target);
void SetSelected(class ACar_TA* Target);
void HandleCarRemoved(class AGameEvent_TA* GameEvent, class ACar_TA* Car);
void HandleReceivedController(class UPlayer* PlayerRef);
void eventConstruct();
void EventSelectedFocusChanged(class UPreMatchLobby_TA* PreMatchLobby);
};

// Class TAGame.CameraState_MenuCamActor_TA
// 0x000C (0x020C - 0x0218)
class UCameraState_MenuCamActor_TA : public UCameraState_CamActor_X
{
public:
    struct FName           CameraTag;          // 0x0210 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CameraState_MenuCamActor_TA");
        }
    }

    return uClassPointer;
};

bool ShouldExecute();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.PremiumGaragePreviewSet_TA
// 0x0070 (0x0060 - 0x00D0)
class UPremiumGaragePreviewSet_TA : public UObject
{
public:
    int32_t           PreviewActorIndices[0x2]; // 0x0060 (0x0008)
    [0x0000004000000001] (CPF_Edit)
    TArray<class ACarPreviewActor_TA*>      CarPreviewActors; // 0x0068
}

```

```
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UOnlineGameParty_TA*           OnlineParty;          // 0x0078 (0x0008)
[0x0000004000002000] (CPF_Transient)
class ACarPreviewActor_TA*          CurrentCar;          // 0x0080 (0x0008)
[0x0000004000000000]
int32_t                            CurrentCarIdx;        // 0x0088 (0x0004)
[0x0000000000000000]
unsigned long                       bFirstSet : 1;       // 0x008C (0x0004)
[0x0000000000000000] [0x00000001]
class UDLCPack_TA*                 QueuedSwapPack;      // 0x0090 (0x0008)
[0x0000000000000000]
int32_t                            QueuedSwapSubIdx;     // 0x0098 (0x0004)
[0x0000000000000000]
struct FScriptDelegate             __EventSwapFinished__Delegate; // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __EventRevealComplete__Delegate; // 0x00B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PremiumGaragePreviewSet_TA");
}

return uClassPointer;
};

void SetPreviewBaseLoadout();
void FinishReveal();
void OnRevealFinished(int32_t SlotIdx);
void OnRevealStarted();
void DoReveal(class UOnlineProduct_TA* OnlineProduct, unsigned long bMultiProductDrop);
void SetupCarReveal(class UProductSlot_TA* RevealSlot);
void InitReveal();
void ResetInvalidAssets(class ACarPreviewActor_TA* PreviewActor, int32_t SlotIndex, int32_t ProductID);
void SetTurnTableYawOffset(int32_t YawOffset);
void SetCarRotations(int32_t Rotation, unsigned long bGamepad);
void ExitPremiumGarage();
void EnterPremiumGarage();
class ATurnTableActor_TA* GetTurntable();
struct FLoadoutData GetCurrentLoadout();
void SetPreviewSlot(class UProductSlot_TA* Slot);
void SetPreviewSlotIndex(int32_t SlotIndex);
void SetPreviewProduct(int32_t SlotIdx, int32_t ProductID, struct FProductInstanceId OnlineID,
int32_t TeamIdx, unsigned long bUsePrimaryBaseLoadout);
void SetOldLoadout(struct FLoadoutData Loadout);
void HandleCarSwapComplete();
void UpdateWheel(struct FAssetLoadResult Result);
void SwapPremiumCarTo(class UDLCPack_TA* DLCPack, int32_t SubIdx);
```

```

void HandleCarPreviewActorAdded(class UGameShare_TA* GameShare, class
ACarPreviewActor_TA* PreviewActor);
void Cleanup();
void Init();
void EventRevealComplete(int32_t SlotIdx);
void EventSwapFinished();
};

// Class TAGame.CameraState_PrePodiumSpotlight_TA
// 0x0014 (0x008C - 0x00A0)
class UCameraState_PrePodiumSpotlight_TA : public UCameraState_X
{
public:
float ElapsedTime; // 0x0090 (0x0004)
[0x0000000000000000] bFadeOut : 1; // 0x0094 (0x0004)
[0x0000000000000000] [0x00000001] float FadeOutDelay; // 0x0098 (0x0004)
[0x0000000000000000] float FadeOutDuration; // 0x009C (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_PrePodiumSpotlight_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldExecute();
};

// Class TAGame.CameraState_ReplayAutoCam_TA
// 0x020C (0x00B4 - 0x02C0)
class UCameraState_ReplayAutoCam_TA : public UCameraState_Replay_TA
{
public:
float FOVInterpSpeed; // 0x00B8 (0x0004)
[0x0000000000000000] float StartViewFocusTime; // 0x00BC (0x0004)
[0x0000000000000000] float FocusPitchInterpSpeed; // 0x00C0 (0x0004)
[0x0000000000000000] float FocusYawInterpSpeed; // 0x00C4 (0x0004)
[0x0000000000000000] float AerialFOV; // 0x00C8 (0x0004)
[0x0000000000000000]

```

float [0x0000000000000000]	CountDownFOV; // 0x00CC (0x0004)
float [0x0000000000000000]	CountDownPanSpeed; // 0x00D0 (0x0004)
float [0x0000000000000000]	CountDownRotateSpeed; // 0x00D4 (0x0004)
float [0x0000000000000000]	CountDownPanBlendAmount; // 0x00D8 (0x0004)
float [0x0000000000000000]	CountDownFocusActorStartTime; // 0x00DC (0x0004)
float [0x0000000000000000]	LocationBlendSpeed; // 0x00E0 (0x0004)
float [0x0000000000000000]	BallAerialZStartTime; // 0x00E4 (0x0004)
float [0x0000000000000000]	MinBallAerialZHeight; // 0x00E8 (0x0004)
float [0x0000000000000000]	MinPlayerAerialZHeight; // 0x00EC (0x0004)
float [0x0000000000000000]	MinFocusActorViewTime; // 0x00F0 (0x0004)
float [0x0000000000000000]	BallFloorFocusPercent; // 0x00F4 (0x0004)
float [0x0000000000000000]	LocationAngleInterpSpeed; // 0x00F8 (0x0004)
float [0x0000000000000000]	LastLocationAngleUpdateTime; // 0x00FC (0x0004)
float [0x0000000000000000]	MinStartRadiusForPlayerAerial; // 0x0100 (0x0004)
float [0x0000000000000000]	MaxStartRadiusForPlayerAerial; // 0x0104 (0x0004)
float [0x0000000000000000]	DesiredCameraDistance; // 0x0108 (0x0004)
float [0x0000000000000000]	ResetToMapDefaultTime; // 0x010C (0x0004)
float [0x0000000000000000]	MaxWaitTimeTillBallHit; // 0x0110 (0x0004)
float [0x0000000000000000]	FocusActorZViewOffset; // 0x0114 (0x0004)
float [0x0000000000000000]	FloorTraceDistance; // 0x0118 (0x0004)
float [0x0000000000000000]	BackAwayFromWallDistance; // 0x011C (0x0004)
float [0x0000000000000000]	NoCarsNearBallDistance; // 0x0120 (0x0004)
float [0x0000000000000000]	MinPlayerCameraFloorHeight; // 0x0124 (0x0004)
float [0x0000000000000000]	FocusPointInterpSpeed; // 0x0128 (0x0004)
float [0x0000000000000000]	MinMaxPitch; // 0x012C (0x0004)
float [0x0000000000000000]	LastDesiredLocationBlendSpeed; // 0x0130 (0x0004)
float [0x0000000000000000]	AdditionalFocusZ; // 0x0134 (0x0004)

```
float TimeSinceIgnoredGoalFocusExtent; // 0x0138 (0x0004)
float LastTimeAPlayerWasNearBall; // 0x013C (0x0004)
float AdditionalBackAwayFromBallSpeed; // 0x0140 (0x0004)
float GoalScorerFocusDistance; // 0x0144 (0x0004)
float MapResetStartDistance; // 0x0148 (0x0004)
float MapResetStartHeight; // 0x014C (0x0004)
float GoalScoredDistance; // 0x0150 (0x0004)
float MaxGoalYZoomOutDistance; // 0x0154 (0x0004)
float MaxGoalXZoomOutDistance; // 0x0158 (0x0004)
float MinCameraDistanceToBall; // 0x015C (0x0004)
unsigned long bSnapFOV : 1; // 0x0160 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long bSnapToFocus : 1; // 0x0160 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long bBallHasBeenHit : 1; // 0x0160 (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long bHasGoalScorerFocus : 1; // 0x0160 (0x0004)
[0x0000000000000000] [0x00000008]
unsigned long bHitWallLastFrame : 1; // 0x0160 (0x0004)
[0x0000000000000000] [0x00000010]
struct FVector BallFloorLocation; // 0x0164 (0x000C)
[0x0000000000000000]
struct FVector BallFloorNormal; // 0x0170 (0x000C)
[0x0000000000000000]
struct FVector LastCarFocusOffset; // 0x017C (0x000C)
[0x0000000000000000]
struct FVector CountDownPanDirection; // 0x0188 (0x000C)
[0x0000000000000000]
struct FVector CountDownPanStartOffset; // 0x0194 (0x000C)
[0x0000000000000000]
struct FVector CountDownRotateOffset; // 0x01A0 (0x000C)
[0x0000000000000000]
struct FVector LastFocusOffset; // 0x01AC (0x000C)
[0x0000000000000000]
struct FVector LastFocusLocation; // 0x01B8 (0x000C)
[0x0000000000000000]
struct FVector LastDesiredLocation; // 0x01C4 (0x000C)
[0x0000000000000000]
struct FRotator LocationAngle; // 0x01D0 (0x000C)
[0x0000000000000000]
struct FRotator DesiredLocationAngle; // 0x01DC (0x000C)
[0x0000000000000000]
struct FRotator DesiredRotation; // 0x01E8 (0x000C)
[0x0000000000000000]
```

```

struct FRotator MapResetStartRotation; // 0x01F4 (0x000C)
[0x0000000000000000]
class APRI_TA* FocusActor; // 0x0200 (0x0008)
[0x0000000000000000]
class APRI_TA* LastScorer; // 0x0208 (0x0008)
[0x0000000000000000]
TArray<class APRI_TA*> AerialPRIs; // 0x0210 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
TArray<class APRI_TA*> CountDownPRIs; // 0x0220 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)
class UGoal_TA* GoalFocus; // 0x0230 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
uint8_t CountDownFocusType; // 0x0238 (0x0001)
[0x0000000000000000]
class ABall_TA* Ball; // 0x0240 (0x0008)
[0x000000000002000] (CPF_Transient)
class AGameEvent_Soccar_TA* SoccarGame; // 0x0248
(0x0008) [0x000000000002000] (CPF_Transient)
class UCameraState_X* ReplayCameraState; // 0x0250 (0x0008)
[0x0000000000000000]
struct FProfileSliderLimits CameraDistanceLimits; // 0x0258 (0x000C)
[0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits CameraZLimits; // 0x0264 (0x000C)
[0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits CameraMoveSpeedLimits; // 0x0270
(0x000C) [0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits CameraMoveSpeedZLimits; // 0x027C
(0x000C) [0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits PlayerCameraDistanceLimits; // 0x0288
(0x000C) [0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits RotateSpeedLimits; // 0x0294 (0x000C)
[0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits PlayerFocusBlendSpeedLimits; // 0x02A0
(0x000C) [0x00000000000003] (CPF>Edit | CPF_Const)
struct FProfileSliderLimits AerialPredictionLimits; // 0x02AC (0x000C)
[0x00000000000003] (CPF>Edit | CPF_Const)
class UProfileCameraSave_TA* CameraSave; // 0x02B8 (0x0008)
[0x000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_ReplayAutoCam_TA");
}

return uClassPointer;
};

void __CameraState_ReplayAutoCam_TA__Init_0x1(class UProfileCameraSave_TA* SO);
void DrawRotatedBox(struct FVector Origin, struct FRotator BoxRotation, struct FVector Extent,

```

```
struct FColor C, unsigned long bPersistent);
bool RotatedBoxContainsPoint(struct FVector Origin, struct FRotator BoxRotation, struct FVector Extent, struct FVector Point);
struct FVector FlattenPointToPlane(struct FVector Center, struct FRotator Rot, struct FVector Point);
class AGameEvent_Soccar_TA* GetSoccarGame();
class UCameraState_X* GetProxyCameraState();
bool LookingAtWall(struct FRotator CameraRotation);
void HandleBallExploded(class ABall_TA* InBall);
void ResetCameraToMapDefault();
float GetDefaultMoveSpeed();
bool WasJustIgnoringGoalFocusExtent(float MinTimeSince);
float ModifyByAerialMultiplier(float Min, float Max);
float GetAerialPredictionMultiplier();
bool GetFocusOnGoals();
bool GetCutToAerialCam();
bool BallIsAboveMinAerialZ();
bool GetAllowFocusActor();
bool GetUseReplayCamera();
float GetFocusActorBlendSpeed();
float GetDesiredAerialActorDistance();
float GetDesiredRotateSpeed();
float GetDesiredMoveSpeedZ();
float GetDesiredMoveSpeed();
float GetDesiredCameraZ();
bool BallHasBeenHit();
void UpdateDesiredCameraDistance(float DeltaTime);
float GetDefaultCameraDistance();
float GetDesiredCameraDistance();
float GetDesiredCameraFOV();
void HandleGameStateChanged(class AGameEvent_TA* GameEvent);
bool ChooseCountDownFocusActor();
void UpdateCountDownTextureStreaming();
class APRI_TA* GetNextEligibleCountDownFocusActor();
bool IsCountDownFocusEligible(class APRI_TA* PRI);
void ResetCountDownPRIs();
void HandleFirstBallHit(class AGameEvent_Soccar_TA* GameEvent);
bool InGameState(struct FName StateName);
class APRI_TA* PredictFirstPlayerToBall();
bool APlayerIsMovingTowardCenterWithinRadius(struct FVector Center, float Radius, float MinDotAngleTowardsBall);
bool MovingTowardsBall(class APRI_TA* PRI, float MaxDotAngle);
bool FlyingTowardsBall(class APRI_TA* PRI, float MinZFlyHeight, float MinAcceptableDistanceToBall);
bool AddToAerialPRIs(class APRI_TA* PRI);
void ResetFocusActorSettings();
class APRI_TA* UpdateAerialPRIs(float PickWaitTime);
bool SetFocusActor(class APRI_TA* PRI);
bool UpdateFocusActor_PostGoal(struct FVector FocusPoint, float DeltaTime, struct FCameraOrientation& OutPOV);
void HandlePlayerScored(class AGameEvent_Soccar_TA* GameEvent, class APRI_TA* Scorer);
void UpdateFocusActor_POV(struct FVector FocusPoint, float DeltaTime, float BlendSpeed,
unsigned long bCutToCamera, struct FCameraOrientation& OutPOV);
bool UpdateFocusActor_Active(struct FVector FocusPoint, float DeltaTime, struct
```

```

FCameraOrientation& OutPOV);
bool UpdateFocusActor_CountDown(struct FVector FocusPoint, float DeltaTime, struct
FCameraOrientation& OutPOV);
bool UpdateFocusActor(struct FVector FocusPoint, float DeltaTime, struct FCameraOrientation&
OutPOV);
void AdjustSideLocation(struct FVector& AdjustedLocation);
bool AdjustDesiredLocationAngleFromHit(struct FVector TraceStart, struct FVector HitLocation,
struct FVector HitNormal, float ExtraTraceDistance, struct FVector& Out_AdjustedLocation);
bool CanFitAtLocation(struct FVector DesiredLocation, float ExtraTraceDistance, struct FVector&
Out_TraceStart, struct FVector& out_HitLocation, struct FVector& out_HitNormal);
void AdjustDesiredLocation(struct FVector& AdjustedLocation);
bool CanMoveCameraToThisAngle(struct FVector CameraLocation, struct FRotator NewAngle,
float DesiredTraceDistance);
struct FRotator GetBallSideAngle(struct FVector CameraLocation);
void SetDesiredLocationAngle(struct FRotator NewAngle);
bool BallToCloseOrMovingTowardCamera(struct FVector CameraLocation);
bool UpdateLocationAngle(struct FCameraOrientation& POV);
struct FVector GetBallFloorFocusPoint(float DeltaTime);
class UGoal_TA* FindGoalWithinRange();
float GetZAngleFromLocation(struct FVector ActorLocation, struct FRotator ActorRotation, struct
FVector TestLocation);
float GetYAngleFromLocation(struct FVector ActorLocation, struct FRotator ActorRotation, struct
FVector TestLocation);
bool FloorTrace(struct FVector TraceStart, float TraceDistance, struct FVector& out_HitLocation,
struct FVector& out_HitNormal);
float GetDesiredZLocation(struct FVector OldCameraLocation, struct FVector
NewCameraLocation, struct FVector DesiredLocation, float& Out_MinZHeight);
void UpdateLocation(struct FVector FocusPoint, float DeltaTime, struct FCameraOrientation&
OutPOV);
void UpdateRotation(struct FVector FocusPoint, float YawInterpSpeed, float PitchInterpSpeed,
float DeltaTime, struct FCameraOrientation& OutPOV);
void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void SetFromMapInfo();
void EndCameraState();
void BeginCameraState();
void Init(class ACamera_X* InCamera);
};


```

```

// Class TAGame.CameraState_ReplayFly_TA
// 0x0004 (0x00B4 - 0x00B8)
class UCameraState_ReplayFly_TA : public UCameraState_Replay_TA
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_ReplayFly_TA");
}
}
```

```

return uClassPointer;
};

};

// Class TAGame.CameraState_ReplayFollow_TA
// 0x0030 (0x00B4 - 0x00E4)
class UCameraState_ReplayFollow_TA : public UCameraState_Replay_TA
{
public:
float FollowDistance; // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinFollowDistance; // 0x00BC (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxFollowDistance; // 0x00C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat FollowRotationRateCurve; // 0x00C8
(0x0018) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bFirstExecution : 1; // 0x00E0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_ReplayFollow_TA");
}

return uClassPointer;
}

void UpdateFocusActorPOV(class AActor* FocusActor, float DeltaTime, struct
FCameraOrientation& OutPOV);
bool IsDisabled(struct FName CameraMode, class FString InFocusActorString);
void eventBeginCameraState();
};

// Class TAGame.CameraState_ReplaySoftAttach_TA
// 0x0020 (0x00B4 - 0x00D4)
class UCameraState_ReplaySoftAttach_TA : public UCameraState_Replay_TA
{
public:
struct FRotator AttachRotation; // 0x00B8 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FVector AttachOffset; // 0x00C4 (0x000C)
[0x0000000000000001] (CPF_Edit)
float MaxDistance; // 0x00D0 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_ReplaySoftAttach_TA");
}

return uClassPointer;
};

void ClampPOV(struct FCameraOrientation& OutPOV);
void UpdateFocusActorPOV(class AActor* FocusActor, float DeltaTime, struct
FCameraOrientation& OutPOV);
bool IsDisabled(struct FName CameraMode, class FString InFocusActorString);
};

// Class TAGame.CameraState_ReplayHardAttach_TA
// 0x0004 (0x00D4 - 0x00D8)
class UCameraState_ReplayHardAttach_TA : public UCameraState_ReplaySoftAttach_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_ReplayHardAttach_TA");
}

return uClassPointer;
};

void UpdateFocusActorPOV(class AActor* FocusActor, float DeltaTime, struct
FCameraOrientation& OutPOV);
bool IsDisabled(struct FName CameraMode, class FString InFocusActorString);
};

// Class TAGame.CameraState_SeasonIntro_TA
// 0x0014 (0x008C - 0x00A0)
class UCameraState_SeasonIntro_TA : public UCameraState_X
{
public:
float [0x0000000000000000] RotateSpeed; // 0x0090 (0x0004)
float [0x0000000000000000] ZHeightPercent; // 0x0094 (0x0004)
float [0x0000000000000000] XOffsetDistance; // 0x0098 (0x0004)
float [0x0000000000000000] DefaultFOV; // 0x009C (0x0004)
};

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_SeasonIntro_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
bool ShouldExecute();
class AGameEvent_Soccar_TA* GetGameEvent();
};

// Class TAGame.CameraState_StadiumSide_TA
// 0x0138 (0x0180 - 0x02B8)
class UCameraState_StadiumSide_TA : public UCameraState_DirectorProxy_TA
{
public:
struct FVector TrackOffset; // 0x0180 (0x000C)
[0x0001000000000001] (CPF_Edit)
struct FVector TrackSize; // 0x018C (0x000C)
[0x0001000000000001] (CPF_Edit)
struct FInterpCurveFloat TrackProximityOffsetSide; // 0x0198
(0x0018) [0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
float TrackProximityOffsetLerp; // 0x01B0 (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FInterpCurveFloat FocusLerp; // 0x01B8 (0x0018)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat TrackLerpSide; // 0x01D0 (0x0018)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat TrackLerpForward; // 0x01E8 (0x0018)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat CarDistWeight; // 0x0200 (0x0018)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat GoalDistWeight; // 0x0218 (0x0018)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)
float BallPredictionTime; // 0x0230 (0x0004)
[0x0001000000000001] (CPF_Edit)
float MinFOV; // 0x0234 (0x0004)
[0x0001000000000001] (CPF_Edit)
float MaxFOV; // 0x0238 (0x0004)
[0x0001000000000001] (CPF_Edit)
float SmoothFOVLerp; // 0x023C (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FVector TrackCenter; // 0x0240 (0x000C)
[0x000100000002000] (CPF_Transient)
struct FVector TrackRight; // 0x024C (0x000C)
[0x000100000002000] (CPF_Transient)

```

```

struct FVector TrackForward; // 0x0258 (0x000C)
[0x0001000000002000] (CPF_Transient)
struct FVector TrackStart; // 0x0264 (0x000C)
[0x0001000000002000] (CPF_Transient)
struct FVector TrackEnd; // 0x0270 (0x000C)
[0x0001000000002000] (CPF_Transient)
struct FVector FieldStart; // 0x027C (0x000C)
[0x0001000000002000] (CPF_Transient)
struct FVector FieldEnd; // 0x0288 (0x000C)
[0x0001000000002000] (CPF_Transient)
struct FVector SmoothedFocus; // 0x0294 (0x000C)
[0x0001000000002000] (CPF_Transient)
struct FVector SmoothedFieldPct; // 0x02A0 (0x000C)
[0x0001000000002000] (CPF_Transient)
float SmoothedFOV; // 0x02AC (0x0004)
[0x0001000000002000] (CPF_Transient)
float SmoothedProximityOffsetSide; // 0x02B0 (0x0004)
[0x0001000000002000] (CPF_Transient)
unsigned long bCut : 1; // 0x02B4 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_StadiumSide_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void InitFieldTrackValues();
void BeginCameraState();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraState_Waiting_TA
// 0x000C (0x020C - 0x0218)
class UCameraState_Waiting_TA : public UCameraState_CamActor_X
{
public:
float MinExecutionTime; // 0x0210 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ExecutionTime; // 0x0214 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraState_Waiting_TA");
}

return uClassPointer;
};

void UpdatePOV(float DeltaTime, struct FCameraOrientation& OutPOV);
void Tick(float DeltaTime);
bool ShouldKeepExecuting();
struct FViewTargetTransitionParams GetEndBlendParams(class UCameraState_X* NewState);
bool ShouldExecute();
};

// Class TAGame.PRI_KnockOut_TA
// 0x00A0 (0x0BD8 - 0x0C78)
class APRI_KnockOut_TA : public APRI_TA
{
public:
class AGameEvent_KnockOut_TA*           GameEvent_KO;           // 0x0BD8
(0x0008) [0x0001004000002000] (CPF_Transient)
unsigned long                          bShowMatchPlacement : 1;      // 0x0BE0 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)
unsigned long                          bIsEliminated : 1;        // 0x0BE0 (0x0004)
[0x0001004100002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long                          bIsActiveMVP : 1;         // 0x0BE0 (0x0004)
[0x0001004100002020] [0x00000004] (CPF_Net | CPF_Transient)
int32_t                               Knockouts;             // 0x0BE4 (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               StackedKnockoutCount;    // 0x0BE8 (0x0004)
[0x0001004000002000] (CPF_Transient)
float                                LastKnockoutTime;       // 0x0BEC (0x0004)
[0x0001004000002000] (CPF_Transient)
float                                MaxKnockoutStackTime;   // 0x0BF0 (0x0004)
[0x0001000000000002] (CPF_Const)
int32_t                               KnockoutAssists;        // 0x0BF4 (0x0004)
[0x0001004000002000] (CPF_Transient)
int32_t                               KnockoutDeaths;         // 0x0BF8 (0x0004)
[0x0001004100002020] (CPF_Net | CPF_Transient)
int32_t                               DamageCaused;          // 0x0BFC (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               Hits;                  // 0x0C00 (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               Grabs;                // 0x0C04 (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               Blocks;               // 0x0C08 (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               EliminationOrder;     // 0x0C0C (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               MatchPlacement;        // 0x0C10 (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
int32_t                               SpectateDelaySeconds; // 0x0C14 (0x0004)

```

```
[0x0001000000000000]
struct FScriptDelegate           __EventKnockedOut__Delegate;           // 0x0C18
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventActiveMVPChanged__Delegate;      // 0x0C30
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventLivesChanged__Delegate;        // 0x0C48
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventEliminated__Delegate;         // 0x0C60
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PRI_KnockOut_TA");
}

return uClassPointer;
};

TArray<struct FScoreboardStat> GetSpectatorStats();
TArray<struct FScoreboardStat> GetScoreboardStats();
bool ShowOnScoreboard();
bool AllowCarRespawn();
bool AllowTeamChange();
int32_t GetLivesRemaining();
bool HasLives();
void SetMatchPlacement(int32_t Placement);
void SetEliminationOrder(int32_t Order);
void OnDisconnect();
void AddGrab();
void AddHit();
void AddBlock();
void AddDamageCaused(int32_t Amount);
void AddKnockoutAssist(class ACar_TA* Victim);
void ResetStackedKnockoutCount();
void AddKnockout(class ACar_TA* Victim);
void OnActiveMVPChanged();
void SetAsActiveMVP(unsigned long bEnable);
void OnLivesChanged();
void SetShowMatchPlacement(unsigned long bInShowMatchPlacement);
void OnEliminated(int32_t SpectateDelay);
bool ShouldChooseTeamWhenSpectating();
void HandleDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void SetCar(class ACar_TA* NewCar);
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void eventReplicatedEvent(struct FName VarName);
void EventEliminated(int32_t SpectateDelay);
void EventLivesChanged(class APRI_KnockOut_TA* PRI);
void EventActiveMVPChanged(class APRI_KnockOut_TA* PRI);
void EventKnockedOut(class APRI_KnockOut_TA* PRI, class ACar_TA* Victim);
```

```

};

// Class TAGame.CameraStateSelector_Priority_TA
// 0x0014 (0x0088 - 0x009C)
class UCameraStateSelector_Priority_TA : public UCameraStateSelector_TA
{
public:
TArray<struct FStateParams> PriorityStates; // 0x0088 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bSelectFirst : 1; // 0x0098 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraStateSelector_Priority_TA");
}

return uClassPointer;
};

class UCameraState_X* SelectState(class UCameraState_X* InCurrentState);
void SetActive();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraStateSelector_Sequence_TA
// 0x0008 (0x0088 - 0x0090)
class UCameraStateSelector_Sequence_TA : public UCameraStateSelector_TA
{
public:
unsigned long bResetIndexOnActive : 1; // 0x0088 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
int32_t StatIdx; // 0x008C (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CameraStateSelector_Sequence_TA");
}

return uClassPointer;
};

int32_t GetNextStateIndex();

```

```
class UCameraState_X* SelectState(class UCameraState_X* InCurrentState);
void SetActive();
};

// Class TAGame.CameraStateSelector_RandomSequence_TA
// 0x0004 (0x0090 - 0x0094)
class UCameraStateSelector_RandomSequence_TA : public
UCameraStateSelector_Sequence_TA
{
public:
unsigned long          bReshuffleOnActive : 1;           // 0x0090 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.CameraStateSelector_RandomSequence_TA");
}

return uClassPointer;
};

void Reshuffle();
void SetActive();
void Init(class ACamera_X* InCamera);
};

// Class TAGame.CameraTrackPoint_TA
// 0x0024 (0x0060 - 0x0084)
class UCameraTrackPoint_TA : public UObject
{
public:
int32_t          frame;           // 0x0060 (0x0004)
[0x0000000000000000]
float            Time;            // 0x0064 (0x0004)
[0x0000000000000000]
struct FVector    Location;        // 0x0068 (0x000C)
[0x0000000000000000]
struct FRotator   Rotation;        // 0x0074 (0x000C)
[0x0000000000000000]
float             FOV;             // 0x0080 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
```

```

{
uClassPointer = UObject::FindClass("Class TAGame.CameraTrackPoint_TA");
}

return uClassPointer;
};

};

// Class TAGame.Cannon_TA
// 0x0041 (0x0268 - 0x02A9)
class ACannon_TA : public AActor
{
public:
class USkeletalMeshComponent* Mesh; // 0x0268 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAnimTree* CannonAnimTree; // 0x0270 (0x0008)
[0x000000000000000]
class UAnimSet* CannonAnimSet; // 0x0278 (0x0008)
[0x000000000000000]
class UAnimNodeSlot* FullBodyAnimSlot; // 0x0280 (0x0008)
[0x000000000000000]
class USkelControlSingleBone* PitchControl; // 0x0288 (0x0008)
[0x000000000000000]
float Pitch; // 0x0290 (0x0004)
[0x0000000100000020] (CPF_Net)
class UParticleSystem* EngineThrusterParticle; // 0x0298 (0x0008)
[0x000000000000000]
class UAkSoundCue* CannonFireSound; // 0x02A0 (0x0008)
[0x000000000000000]
uint8_t FireCount; // 0x02A8 (0x0001)
[0x0000000100000020] (CPF_Net)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Cannon_TA");
}

return uClassPointer;
};

void Fire();
void DoSimulatedFire();
void PlayFullBodyAnim(struct FName AnimName, float Rate, float BlendIn, float BlendOut,
unsigned long bLoop);
void SetPitch(float InPitch);
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
};

```

```

// Class TAGame.LocalPlayerAudioParamsComponent_TA
// 0x0018 (0x0070 - 0x0088)
class ULocalPlayerAudioParamsComponent_TA : public UComponent
{
public:
    class UISetParameter*           ParameterInterface_Object;      // 0x0070
    (0x0008) [0x0000004000000000]
    class UISetParameter*           ParameterInterface_Interface;   // 0x0078
    (0x0008) [0x0000004000000000]
    class APlayerController*        PC;                                // 0x0080 (0x0008)
    [0x0000004000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.LocalPlayerAudioParamsComponent_TA");
        }

        return uClassPointer;
    }

void __LocalPlayerAudioParamsComponent_TA__Init_0x3(class AGameEvent_TA* _);
void __LocalPlayerAudioParamsComponent_TA__Init_0x2(class ACamera_TA* C, class AActor* T);
void __LocalPlayerAudioParamsComponent_TA__Init_0x1(class ACamera_X* C, class UCameraState_X* CS);
void UpdateReplayCameraLocalView(class ACamera_Replay_TA* Camera);
void UpdateNumLocalPlayers(class UGameViewportClient_TA* GVC, int32_t LocalPlayerCount);
void HandleLocalPlayerChange(class ULocalPlayer* NewPlayer);
int32_t GetListenerFromSplitScreenPosition(uint8_t InSplitScreenPosition);
void ApplyParams(unsigned long bIsLocal, uint8_t SplitScreenPosition);
void UpdateParams();
void Init(class APlayerController* InPC, class UISetParameter* InParameterInterface);
};

// Class TAGame.TeamColorPreferences_TA
// 0x0020 (0x0060 - 0x0080)
class UTeamColorPreferences_TA : public UObject
{
public:
    unsigned long                 bColorBlind : 1;                  // 0x0060 (0x0004)
    [0x0000000000000000] [0x00000001]
    unsigned long                 bForceDefaultColors : 1;        // 0x0060 (0x0004)
    [0x0000000000000000] [0x00000002]
    struct FScriptDelegate        EventPropertyChange;             // 0x0068 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TeamColorPreferences_TA");
}

return uClassPointer;
};

void EventPropertyChangeFunc();
bool AllowCustomTeamColors();
};

// Class TAGame.ThrottleStateComponent_TA
// 0x0024 (0x00A4 - 0x00C8)
class UThrottleStateComponent_TA : public UActorComponent_X
{
public:
class UAkSoundCue*          EngageSound;           // 0x00A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*          ReleaseSound;          // 0x00B0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AVehicle_TA*          Car;                   // 0x00B8 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                        PrevThrottle;         // 0x00C0 (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long                bPrevCarAccelerating : 1; // 0x00C4 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long                bPrevCarIdling : 1;    // 0x00C4 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
unsigned long                bCanThrottleEngage : 1; // 0x00C4 (0x0004)
[0x0000000000002000] [0x00000004] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ThrottleStateComponent_TA");
}

return uClassPointer;
};

void StopThrottleSounds();
void eventTick(float DeltaTime);
void eventDetached();
void eventAttached();
};

```

```

// Class TAGame.CarTrajectoryComponent_TA
// 0x0058 (0x00A4 - 0x00FC)
class UCarTrajectoryComponent_TA : public UActorComponent_X
{
public:
    unsigned long          bEnabled : 1;           // 0x00A8 (0x0004)
    [0x0000004000002000] [0x00000001] (CPF_Transient)
    TArray<struct FVector> Points;             // 0x00B0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t                TotalDrawPoints;      // 0x00C0 (0x0004)
    [0x0000000000000002] (CPF_Const)
    int32_t                FramesBetweenPoints; // 0x00C4 (0x0004)
    [0x0000000000000002] (CPF_Const)
    float                  LastAngle;           // 0x00C8 (0x0004)
    [0x0000000000002000] (CPF_Transient)
    struct FRotator         VelocityRot;        // 0x00CC (0x000C)
    [0x0000000000002000] (CPF_Transient)
    struct FVector          CurrentVelocity;   // 0x00D8 (0x000C)
    [0x0000000000002000] (CPF_Transient)
    struct FVector          LastRotateAxis;    // 0x00E4 (0x000C)
    [0x0000000000002000] (CPF_Transient)
    struct FVector          LastVelocity;       // 0x00F0 (0x000C)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CarTrajectoryComponent_TA");
        }

        return uClassPointer;
    };

    float GetPointInterpSpeed(int32_t PointIndex);
    void eventTick(float DeltaTime);
    void eventAttached();
    void SetEnabled(unsigned long bEnable);
};

// Class TAGame.NameplateComponentCar_TA
// 0x0008 (0x00E0 - 0x00E8)
class UNameplateComponentCar_TA : public UNameplateComponent_TA
{
public:
    class ACar_TA*          Car;               // 0x00E0 (0x0008)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NameplateComponentCar_TA");
}

return uClassPointer;
};

int32_t GetSpectatorShortcut();
bool IsReady();
bool IsLocalPlayer(class APRI_TA* LocalPRI);
int32_t GetPlayerTeam();
class FString GetPlayerName();
class APRI_TA* GetPRI();
void eventAttached();
};

// Class TAGame.CarDemoActor_TA
// 0x0008 (0x0268 - 0x0270)
class ACarDemoActor_TA : public AActor
{
public:
class UStaticMeshComponent*           StaticMeshComponent;          // 0x0268
(0x0008) [0x0000000040A000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_EditConst |
CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarDemoActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.DemoExplosionHandler_TA
// 0x0008 (0x0150 - 0x0158)
class UDemoExplosionHandler_TA : public UExplosionHitHandler_TA
{
public:
class AFXActor_X*           DemoFXToApply;          // 0x0150 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DemoExplosionHandler_TA");
}

return uClassPointer;
};

void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

// Class TAGame.DemolishedCar_TA
// 0x0028 (0x0060 - 0x0088)
class UDemolishedCar_TA : public UObject
{
public:
struct FDemolishData           Data;                      // 0x0060 (0x0028)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DemolishedCar_TA");
}

return uClassPointer;
};

};

// Class TAGame.Car_Freeplay_TA
// 0x0010 (0xB4860 - 0xB5870)
class ACar_Freeplay_TA : public ACar_TA
{
public:
TArray<struct FLinearColor>      CarColors;            // 0xB4860 (0x0010)
[0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Car_Freeplay_TA");
}
}

```

```

return uClassPointer;
};

static void SetCarColor(class UCarMeshComponentBase_TA* InMesh, unsigned long
bSetColor, unsigned long bSetCustomColor);
void HandleAllAssetsLoaded(class UProductLoader_TA* Loader);
int32_t GetLoadoutTeamIndex();
bool HasTeam();
void InitFX();
};

// Class TAGame.FXActor_Knockout_Attack_TA
// 0x0018 (0x03678 - 0x03890)
class AFXActor_Knockout_Attack_TA : public AFXActor_TA
{
public:
    uint8_t AttackType; // 0x03678 (0x0001)
    [0x0001000000002000] (CPF_Transient)
    TArray<class UFXActorEvent_X*> AttackEvents; // 0x03780
    (0x0010) [0x000100000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FXActor_Knockout_Attack_TA");
        }
    }

    return uClassPointer;
};

void __FXActor_Knockout_Attack_TA__UpdateAttackType_0x1(class UFXActorEvent_X* E);
void UpdateAttackType(uint8_t NewAttackType);
};

// Class TAGame.Stunlock_TA
// 0x0058 (0x0268 - 0x02C0)
class ASTunlock_TA : public AActor
{
public:
    class ACar_KnockOut_TA* Car; // 0x0268 (0x0008)
    [0x0001004100002020] (CPF_Net | CPF_Transient)
    float ButtonMashReduceTime; // 0x0270 (0x0004)
    [0x0001000000000002] (CPF_Const)
    float MinTimeBetweenMashes; // 0x0274 (0x0004)
    [0x0001000000000002] (CPF_Const)
    float LastMashTime; // 0x0278 (0x0004)
    [0x0001000000002000] (CPF_Transient)
    float MaxStunTime; // 0x027C (0x0004)
    [0x0001000100002020] (CPF_Net | CPF_Transient)

```

```
float StunTimeRemaining; // 0x0280 (0x0004)
[0x0001000000002020] (CPF_Net | CPF_Transient)
float MashTime; // 0x0284 (0x0004)
[0x0001004000002020] (CPF_Net | CPF_Transient)
unsigned long bMashed : 1; // 0x0288 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)
struct FScriptDelegate __EventStunlockComplete__Delegate; // 0x0290
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventStunlockTimeChanged__Delegate; // 0x02A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Stunlock_TA");
}

return uClassPointer;
};

void SetComplete();
void eventTick(float DeltaTime);
void TimeUnmash();
void TimeMash();
float GetMashTimeAlpha();
float GetTimeRemainingAlpha();
float GetTimeRemaining();
void Start(float InStunTime, float InMashTime);
void OnCarSet();
void SetCar(class ACar_KnockOut_TA* InCar);
void eventReplicatedEvent(struct FName VarName);
void EventStunlockTimeChanged(class AStunlock_TA* Stunlock);
void EventStunlockComplete();
};

// Class TAGame.Car_Season_TA
// 0x0000 (0xB4860 - 0xB4860)
class ACar_Season_TA : public ACar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Car_Season_TA");
}
}
```

```

return uClassPointer;
};

int32_t GetLoadoutTeamIndex();
void HandleLogoChanged(class ATeam_TA* Team);
void UpdateTeamLogo();
bool UpdateTeamLoadout();
};

// Class TAGame.ProfileControlsSave_TA
// 0x0048 (0x00C8 - 0x0110)
class UProfileControlsSave_TA : public USaveObject_TA
{
public:
    float           WaveformScalar;          // 0x00C8 (0x0004)
    [0x0000000000000002] (CPF_Const)
    uint8_t          ForceFeedbackType;       // 0x00CC (0x0001)
    [0x0000000000000000]
    uint8_t          ForceFeedbackMode;        // 0x00CD (0x0001)
    [0x0000000000000000]
    float           LegacyForceFeedbackScales[0x6]; // 0x00D0 (0x0018)
    [0x0000000000002002] (CPF_Const | CPF_Transient)
    float           ForceFeedbackScale;        // 0x00E8 (0x0004)
    [0x0000000000000000]
    unsigned long    bPushToTalk : 1;           // 0x00EC (0x0004)
    [0x0000000000000000] [0x00000001]
    struct FName     ControlPresetName;        // 0x00F0 (0x0008)
    [0x0008000000000000]
    struct FScriptDelegate __ControlPresetName__ChangeNotify; // 0x00F8
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProfileControlsSave_TA");
        }
    }

    return uClassPointer;
};

void __ProfileControlsSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void __ControlPresetName__ChangeNotifyFunc();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
float GetWaveformScale();
void OnCreate();
};

// Class TAGame.CarJointSpinner_TA
// 0x0038 (0x00A4 - 0x00DC)

```

```

class UCarJointSpinner_TA : public UActorComponent_X
{
public:
struct FName           ControlName;          // 0x00A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                 IdleSpeed;           // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                 DrivingSpeed;        // 0x00B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                 BoostSpeed;          // 0x00B8 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector         RotationAxis;        // 0x00BC (0x000C)
[0x0000000000000001] (CPF_Edit)
class ACar_TA*         Car;                // 0x00C8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class USkelControlSingleBone*   Control;    // 0x00D0 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                 Speed;               // 0x00D8 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarJointSpinner_TA");
}

return uClassPointer;
};

void eventTick(float DeltaTime);
void eventAttached();
};

// Class TAGame.CarMeshComponentHelper
// 0x0000 (0x0060 - 0x0060)
class UCarMeshComponentHelper : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarMeshComponentHelper");
}

return uClassPointer;
};

```

```

};

void AttachFXActor(class AFXActor_X* FXArchetype);
};

// Class TAGame.CarPreviewAnim_TA
// 0x0084 (0x0A4 - 0x128)
class UCarPreviewAnim_TA : public UActorComponent_X
{
public:
    class UProductSlot_TA*           Slot;           // 0xA8 (0x008)
    [0x0000000000000001] (CPF_Edit)
    TArray<struct FPreviewAnimKeyframe> KeyFrames; // 0xB0 (0x00B0)
    (0x0010) [0x000000000001] (CPF_Edit | CPF_NeedCtorLink)
    float               Delay;          // 0xC0 (0x004)
    [0x0000000000000001] (CPF_Edit)
    unsigned long        bDelayHidden : 1; // 0xC4 (0x004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long        bLoop : 1;      // 0xC4 (0x004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    unsigned long        bComponentWasAttached : 1; // 0xC4 (0x004)
    [0x0000000000002000] [0x00000004] (CPF_Transient)
    struct FName          SkelControlName; // 0xC8 (0x008)
    [0x0000000000000001] (CPF_Edit)
    class ACarPreviewActor_TA* PreviewActor; // 0xD0 (0x008)
    [0x0000004000002000] (CPF_Transient)
    class UPrimitiveComponent* Component; // 0xD8 (0x008)
    [0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
    struct FPreviewAnimKeyframe FinalKeyframe; // 0xE0 (0x038)
    [0x0000004000002000] (CPF_Transient)
    float                KeyframeTime; // 0x118 (0x004)
    [0x0000004000002000] (CPF_Transient)
    int32_t              KeyframeIndex; // 0x11C (0x004)
    [0x0000004000002000] (CPF_Transient)
    class USkelControlSingleBone* SkelControl; // 0x120 (0x008)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CarPreviewAnim_TA");
        }

        return uClassPointer;
    }

    void SetRotation(struct FRotator Rotation);
    void SetTranslation(struct FVector Translation);
    void UpdateAnim();
    void UpdateKeyframeIndex(float DeltaTime);
}

```

```

void eventTick(float DeltaTime);
void eventDetached();
void Start(class ACarPreviewActor_TA* InOwner, class UPrimitiveComponent* InComponent);
};

// Class TAGame.ProductAttribute_PreviewWheelSpin_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_PreviewWheelSpin_TA : public UProductAttribute_TA
{
public:
float PreviewFrontWheelSpinSpeed; // 0x0080 (0x0004)
[0x0000000000000001] (CPF_Edit)
float PreviewBackWheelSpinSpeed; // 0x0084 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_PreviewWheelSpin_TA");
}

return uClassPointer;
};

};

// Class TAGame.EngineAudioPreviewRev_TA
// 0x0024 (0x0084 - 0x00A8)
class UEngineAudioPreviewRev_TA : public UEngineAudioPreviewBase_TA
{
public:
class UEngineAudioProfileREV_TA* Profile; // 0x0088 (0x0008)
[0x0000000000000000]
class UAkRevPhysicsSimulation* PhysicsSim; // 0x0090 (0x0008)
[0x0000000000000000]
struct FAkRevSimFrame SimFrame; // 0x0098 (0x0010)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngineAudioPreviewRev_TA");
}

return uClassPointer;
};

```

```

void __EngineAudioPreviewRev_TA__Init_0x1(class UAkRevPhysicsSimulation* S);
void Init(class UProductAsset_EngineAudio_TA* InAsset, class UAkParamGroup* InAkParams);
};

// Class TAGame.RotateComponent_TA
// 0x00C4 (0x00A4 - 0x0168)
class URotateComponent_TA : public UActorComponent_X
{
public:
float Momentum; // 0x00A8 (0x0004)
[0x0000000000000000] LastMomentum; // 0x00AC (0x0004)
float RotationSpeed; // 0x00B0 (0x0004)
[0x0000000000000000] struct FRotator CurrentRotation; // 0x00B4 (0x000C)
[0x0000040000000000] float TurntableDirection; // 0x00C0 (0x0004)
[0x0000000000002000] (CPF_Transient) class UAkSoundCue* SFX_Rotating; // 0x00C8 (0x0008)
[0x0000000000000001] (CPF_Edit) class UAkSoundCue* SFX_Reversing; // 0x00D0 (0x0008)
[0x0000000000000001] (CPF_Edit) struct FInterpCurveFloat FallOffCurve; // 0x00D8 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink) struct FInterpCurveFloat YawSpeedCurveMouse; // 0x00F0
(0x0018) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink) struct FInterpCurveFloat YawSpeedCurveGamepad; // 0x0108
(0x0018) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink) unsigned long bGrabbed : 1; // 0x0120 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient) unsigned long bDeterminedSnapRotation : 1; // 0x0120 (0x0004)
[0x0000000000000000] [0x00000002] unsigned long bUsingGamepad : 1; // 0x0120 (0x0004)
[0x0000000000000000] [0x00000004] struct FRotator SnapRotation; // 0x0124 (0x000C)
[0x0000000000000000] float SnapAngle; // 0x0130 (0x0004)
int32_t TotalSnaps; // 0x0134 (0x0004)
struct FRotator SnapStartRotation; // 0x0138 (0x000C)
float MinStopMomentum; // 0x0144 (0x0004)
float SnapBlendSpeed; // 0x0148 (0x0004)
float Input; // 0x014C (0x0004)
struct FScriptDelegate __EventRotationChanged__Delegate; // 0x0150
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RotateComponent_TA");
}

return uClassPointer;
};

void DontStartSound();
void DontPlayReverse();
bool IsTimerActive(struct FName FuncName);
void SetTimer(float Time, unsigned long bLoop, struct FName FuncName);
void ClearTimer(struct FName FuncName);
float CalculateMomentumFalloff(float DeltaTime);
void SetGrabbed(unsigned long bGrab);
float DetermineTurnSpeed(float DeltaTime, float YawInput, unsigned long bGamepad);
void DetermineSnapRotation(int32_t MoveDirection);
void UpdateSpin(float DeltaTime, float YawInput, unsigned long bGamepad);
static int32_t IncrementSlotIndex(int32_t InTotalSnaps, int32_t CurrentIndex, int32_t AddAmount);
struct FRotator GetSlotRotation(int32_t SlotIndex);
static int32_t GetClosestSlotInRotation(struct FVector WorldDirection, struct FRotator InRotation,
int32_t InTotalSnaps, float InSnapAngle);
void SetSnap(struct FRotator InSnapStartRotation, int32_t InTotalSnaps, float InSnapAngle, float InMinStopMomentum);
void StopMovement();
void StopRotatingSound();
void SetComponentRotation(float InDirectionMag, float DeltaTime);
void SetCurrentRotation(struct FRotator NewRotation);
void ResetInput();
void SetInput(float YawInput, unsigned long bGamepad);
void eventTick(float DeltaTime);
void eventBeginPlay();
void EventRotationChanged(class URotateComponent_TA* RotateComponent, struct FRotator NewRotation);
};

// Class TAGame.PlayerController_Menu_TA
// 0x0048 (0x0980 - 0x09C8)
class APlayerController_Menu_TA : public APlayerControllerBase_TA
{
public:
TArray<class UPlayerInputSequence_TA*> InputSequences; // 0x0980
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long bExecutedKonamiCode : 1; // 0x0990 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
struct FScriptDelegate __EventKonamiCode__Delegate; // 0x0998
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventBecomePrimaryPlayer__Delegate; // 0x09B0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerController_Menu_TA");
}

return uClassPointer;
};

void ShowControllerApplet();
void ShowAccountPicker();
void MusicTogglePlaylistSelection();
void MusicNextTrack();
void KonamiCode();
void eventPlayerTick(float DeltaTime);
void ClientSetOnlineStatus();
void RemoveInputSequence(struct FName SequenceName);
void RemoveSplashScreenSequences();
void eventDestroyed();
int32_t GetInputSequencePriority(class UPlayerInputSequence_TA* Sequence);
void HandleCameraSave(class UProfileCameraSave_TA* CameraSettings);
void eventReceivedPlayer();
void EventBecomePrimaryPlayer(class APlayerController_Menu_TA* Primary);
void EventKonamiCode();
};

// Class TAGame.PartyMemberLoadout_TA
// 0x00B1 (0x0060 - 0x0111)
class UPartyMemberLoadout_TA : public UObject
{
public:
struct FUniqueNetId           Uniqueld;          // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FLoadoutData           Loadout;           // 0x00A8 (0x0040)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData>   OnlineLoadoutData;    // 0x00E8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
int32_t                         GaragePreviewTeam; // 0x00F8 (0x0004)
[0x0000000000000000]
class ULocalPlayer*             LocalPlayer;        // 0x0100 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UOnlineProductStoreSet_TA* ProductSet;        // 0x0108 (0x0008)
[0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
uint8_t                           SortWeight;         // 0x0110 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMemberLoadout_TA");
}

return uClassPointer;
};

struct FOnlineProductData GetAvatarBorderData();
struct FOnlineProductData GetBannerData();
};

// Class TAGame.CarThumbnailActor_TA
// 0x0008 (0x0268 - 0x0270)
class ACarThumbnailActor_TA : public AActor
{
public:
class UCarMeshComponent_TA*           Mesh;           // 0x0268 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CarThumbnailActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.ChallengeRequirement_TA
// 0x001C (0x0080 - 0x009C)
class UChallengeRequirement_TA : public UListenObject_TA
{
public:
class FString                  DisplayName;      // 0x0080 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t                      RequiredCount;    // 0x0090 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                      ProgressCount;    // 0x0094 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                      ProgressChange;   // 0x0098 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeRequirement_TA");
}

return uClassPointer;
};

bool ProgressHasChanged();
void ResetChangedProgress();
void SetProgress(struct FChallengeRequirementProgress& Progress);
};

// Class TAGame.GFxData_Notification_TA
// 0x0004 (0x0094 - 0x0098)
class UGFxData_Notification_TA : public UGFxDataRow_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Notification_TA");
}

return uClassPointer;
};

class UNotification_TA* GetNotification();
void SetShown(unsigned long bValue);
void ClickDeny();
void ClickAccept();
void DismissNotification();
void ClearPopUp();
void HandleRemoveNotification(class UNotification_TA* Notification);
void SetNotification(class UNotification_TA* Notification);
};

// Class TAGame.ChallengeNotification_TA
// 0x0014 (0x0170 - 0x0184)
class UChallengeNotification_TA : public UNotification_TA
{
public:
class UTexture*           Icon;           // 0x0170 (0x0008)
[0x0001000040002000] (CPF_Transient | CPF_EditInlineNotify)
class UTexture*           NotifyImage;    // 0x0178 (0x0008)
[0x0001000040002000] (CPF_Transient | CPF_EditInlineNotify)
int32_t                   ChallengeID;   // 0x0180 (0x0004)

```

[0x0001000040000000] (CPF\_EditInlineNotify)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ChallengeNotification_TA");  
}  
  
return uClassPointer;  
};  
  
void RemoveNotification();  
void HandleIconChanged(class UChallengeDefault_TA* Challenge);  
class UChallengeNotification_TA* SetChallenge(class UChallengeDefault_TA* Challenge);  
};  
  
// Class TAGame.ChallengeDefaultManager_TA  
// 0x0080 (0x0060 - 0x00E0)  
class UChallengeDefaultManager_TA : public UObject  
{  
public:  
TArray<class UChallengeDefault_TA*> DefaultChallenges; // 0x0060  
(0x0010) [0x0001004000400000] (CPF_NeedCtorLink)  
TArray<class UChallengeFolder_TA*> Folders; // 0x0070 (0x0010)  
[0x0001004000400000] (CPF_NeedCtorLink)  
class UChallengeConfig_TA* ChallengeConfig; // 0x0080 (0x0008)  
[0x0001800000000000]  
TArray<class UPsyNetService_ChallengeDefault_TA*> NewChallengesNotifications; //  
0x0088 (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)  
struct FScriptDelegate __EventChallengeAdded__Delegate; // 0x0098  
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)  
struct FScriptDelegate __EventChallengesSynced__Delegate; // 0x00B0  
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)  
struct FScriptDelegate __EventChallengesPushed__Delegate; // 0x00C8  
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ChallengeDefaultManager_TA");  
}  
  
return uClassPointer;  
};  
  
bool __ChallengeDefaultManager_TA__ProcessChallengesPushedNotifications_0x2(int32_t C);
```

```

int32_t __ChallengeDefaultManager_TA__ProcessChallengesPushedNotifications_0x1(class UChallengeDefault_TA* C);
void __ChallengeDefaultManager_TA__AddDefaultChallenges_0x1(class UChallengeFolder_TA* F);
void __ChallengeDefaultManager_TA__Cheat_SetFolders_0x1(class UChallengeFolder_TA* F);
void Cheat_SetFolders(TArray<class UChallengeFolder_TA*>& InFolders);
TArray<class UChallengeFolder_TA*> GetSubFolders(class FString CodeName);
class UChallengeFolder_TA* FindFolder(class FString& FindCodeName);
int32_t FindChallengeIndex(int32_t Id);
void RemoveDefaultChallenges();
void RemoveDefaultChallenge(int32_t Id);
void AddDefaultChallenge(class UChallengeDefault_TA* ChallengeDefault);
void AddDefaultChallenges(TArray<class UChallengeDefault_TA*>& InChallenges, TArray<class UChallengeFolder_TA*>& InFolders);
void ProcessChallengesPushedNotifications();
void HandleChallengesPushed(class UPsyNetService_ChallengeDefault_TA* Message);
void HandleChallengesSynced(class URPC_GetChallengeDefault_TA* RPC);
void ResyncChallenges();
void HandlePsyNetLogin();
void eventConstruct();
void EventChallengesPushed(TArray<int32_t> NewChallengeIDs, TArray<int32_t> RemovedChallengeIDs);
void EventChallengesSynced(class UChallengeDefaultManager_TA* Manager);
void EventChallengeAdded(class UChallengeDefaultManager_TA* Manager, class UChallengeDefault_TA* ChallengeDefault);
};

// Class TAGame.ChallengeAvailableNotification_TA
// 0x0004 (0x0184 - 0x0188)
class UChallengeAvailableNotification_TA : public UChallengeNotification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeAvailableNotification_TA");
}

return uClassPointer;
};

void __ChallengeAvailableNotification_TA__SetChallenge_0x1(class UChallengeDefault_TA* C);
class UChallengeNotification_TA* SetChallenge(class UChallengeDefault_TA* Challenge);
};

// Class TAGame.GFxData_ChallengeAvailableNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeAvailableNotification_TA : public UGFxData_Notification_TA
{

```

```
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ChallengeAvailableNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.ChallengeCompleteNotification_TA
// 0x0004 (0x0184 - 0x0188)
class UChallengeCompleteNotification_TA : public UChallengeNotification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeCompleteNotification_TA");
}

return uClassPointer;
};

void __ChallengeCompleteNotification_TA__SetChallenge_0x1(class UChallengeDefault_TA* C);
class UChallengeNotification_TA* SetChallenge(class UChallengeDefault_TA* Challenge);
};

// Class TAGame.GFxData_ChallengeCompleteNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeCompleteNotification_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ChallengeCompleteNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.ChallengeConfig_TA
// 0x0044 (0x0078 - 0x00BC)
class UChallengeConfig_TA : public UOnlineConfig_X
{
public:
int32_t MinSecondsDelayNewNotification; // 0x0078 (0x0004)
[0x0001000000000000]
int32_t MaxSecondsDelayNewNotification; // 0x007C (0x0004)
[0x0001000000000000]
int32_t SecondsDelayNewNotification; // 0x0080 (0x0004)
[0x0001004000002000] (CPF_Transient)
class FString TrainingFolderName; // 0x0088 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString WeeklyFolderName; // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString SeasonFolderName; // 0x00A8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bHideChallengeUI : 1; // 0x00B8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeConfig_TA");
}

return uClassPointer;
};

void Apply();
};

// Class TAGame.PsyNetService_ChallengeDefault_TA
// 0x0030 (0x0090 - 0x00C0)
class UPsyNetService_ChallengeDefault_TA : public UPsyNetClientService_X
{
public:
TArray<class UChallengeFolder_TA*> Folders; // 0x0090 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<class UChallengeDefault_TA*> Challenges; // 0x00A0
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

```

```

TArray<int32_t>           ChallengesIDsToRemove;           // 0x00B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ChallengeDefault_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_GetChallengeDefault_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_GetChallengeDefault_TA : public URPC_X
{
public:
TArray<class UChallengeDefault_TA*>      Challenges;           // 0x00E8
(0x0010) [0x001004000400000] (CPF_NeedCtorLink)
TArray<class UChallengeFolder_TA*>      Folders;             // 0x00F8 (0x0010)
[0x001004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetChallengeDefault_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ItemDropGroup_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ItemDropGroup_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ItemDropGroup_TA");
}

return uClassPointer;
};

bool IsValid();
uint8_t GetDisplayOrder();
};

// Class TAGame.ChallengeDropGroup_TA
// 0x0008 (0x00DC - 0x00E4)
class UChallengeDropGroup_TA : public UCompleteTaskDropGroup_TA
{
public:
unsigned long           bPremium : 1;           // 0x00E0 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeDropGroup_TA");
}

return uClassPointer;
};

void InitFromChallenge(class UChallenge_TA* Challenge);
};

// Class TAGame.GFxData_ChallengeDropGroup_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeDropGroup_TA : public UGFxData_ItemDropGroup_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeDropGroup_TA");
}

return uClassPointer;
};

```

```
};

// Class TAGame.ChallengeEvents_TA
// 0x0018 (0x0060 - 0x0078)
class UChallengeEvents_TA : public UObject
{
public:
struct FScriptDelegate           __EventChallengesChanged__Delegate;      // 0x0060
(0x0018) [0x00000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeEvents_TA");
}

return uClassPointer;
};

void EventChallengesChanged();
};

// Class TAGame.ChallengeFilter_TA
// 0x0000 (0x0060 - 0x0060)
class UChallengeFilter_TA : public UObject
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeFilter_TA");
}

return uClassPointer;
};

TArray<class UChallenge_TA*> FilterChallenges(TArray<class UChallenge_TA*>& AllChallenges);
};

// Class TAGame.ChallengeFilter_Default_TA
// 0x0000 (0x0060 - 0x0060)
class UChallengeFilter_Default_TA : public UChallengeFilter_TA
{
public:
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeFilter_Default_TA");
}

return uClassPointer;
};

TArray<class UChallenge_TA*> FilterChallenges(TArray<class UChallenge_TA*>& AllChallenges);
};

// Class TAGame.ChallengeFilter_Progress_TA
// 0x0000 (0x0060 - 0x0060)
class UChallengeFilter_Progress_TA : public UChallengeFilter_Default_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChallengeFilter_Progress_TA");
}

return uClassPointer;
};

TArray<class UChallenge_TA*> FilterChallenges(TArray<class UChallenge_TA*>& AllChallenges);
};

// Class TAGame.PsyNetService_ChallengeProgress_TA
// 0x0010 (0x0090 - 0x00A0)
class UPsyNetService_ChallengeProgress_TA : public UPsyNetClientService_X
{
public:
TArray<struct FChallengeProgress> Challenges; // 0x0090 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ChallengeProgress_TA");
}

return uClassPointer;
};

};

// Class TAGame.Tutorial_TA
// 0x01BC (0x008C - 0x0248)
class UTutorial_TA : public UTickableStateObject_X
{
public:
    class USequenceAction*           KNode_Owner;          // 0x0090 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class APlayerController_TA*      PC;                  // 0x0098 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class AGameEvent_Tutorial_TA*   GameEvent;           // 0x00A0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class FString                   ModalSectionName;    // 0x00A8 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class UGFxModal_X*             MessageModal;        // 0x00B8 (0x0008)
    [0x00000000002000] (CPF_Transient)
    TArray<struct FMessageInfo>   Messages;            // 0x00C0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t                         MessageIndex;        // 0x00D0 (0x0004)
    [0x0000000000000000]
    float                            DisplayMessageStartTime; // 0x00D4 (0x0004)
    [0x0000000000000000]
    unsigned long                    bEndTutorialOnFinalMessage : 1; // 0x00D8 (0x0004)
    [0x0000000000000001] [0x00000001]
    unsigned long                    bCompleteTutorialOnFinalMessage : 1; // 0x00D8
    (0x0004) [0x0000000000000000] [0x00000002]
    unsigned long                    bInitialMessagesFinished : 1; // 0x00D8 (0x0004)
    [0x0000000000000000] [0x00000004]
    unsigned long                    bFadeAudio : 1;        // 0x00D8 (0x0004)
    [0x0000000000000008]
    unsigned long                    bSkipFadeIn : 1;       // 0x00D8 (0x0004)
    [0x0000000000000010]
    unsigned long                    bFadeTransitionComplete : 1; // 0x00D8 (0x0004)
    [0x0000000000000020]
    unsigned long                    bTutorialStarted : 1; // 0x00D8 (0x0004)
    [0x0000000000000040]
    unsigned long                    bActionNameToDetectPressed : 1; // 0x00D8
    (0x0004) [0x0000000000000000] [0x00000080]
    unsigned long                    bCarTouched : 1;        // 0x00D8 (0x0004)
    [0x0000000000000000] [0x00000100]
    unsigned long                    bCompleteOnBallTouch : 1; // 0x00D8 (0x0004)
    [0x0000000000000000] [0x00000200]
    unsigned long                    bFailTutorialOnOtherCarHitBall : 1; // 0x00D8 (0x0004)
    [0x0000000000000000] [0x00000400]
    unsigned long                    bDestroyBallsOnFadeIn : 1; // 0x00D8 (0x0004)
    [0x0000000000000000] [0x00000800]
    unsigned long                    bDestroyBotsOnFadeIn : 1; // 0x00D8 (0x0004)

```

```
[0x0000000000000000] [0x00001000]                                // 0x00D8 (0x0004)
unsigned long          bHasBot : 1;                                // 0x00D8 (0x0004)
[0x0000000000000000] [0x00002000]
unsigned long          bFoundBot : 1;                                // 0x00D8 (0x0004)
[0x0000000000000000] [0x00004000]
unsigned long          bStartWithSecondaryCamActive : 1;        // 0x00D8
(0x0004) [0x0000000000000000] [0x00008000]
unsigned long          bStartWithBoostMeterDisabled : 1;        // 0x00D8
(0x0004) [0x0000000000000000] [0x00010000]
unsigned long          bBotCanPlay : 1;                                // 0x00D8 (0x0004)
[0x0000000000000000] [0x00020000]
unsigned long          bStartBotOnAllMessagesDisplayed : 1;      // 0x00D8
(0x0004) [0x0000000000000000] [0x00040000]
unsigned long          bBallCanHitPlayersGoal : 1;                // 0x00D8 (0x0004)
[0x0000000000000000] [0x00080000]
unsigned long          bCompleteOnLastBallTouched : 1;            // 0x00D8
(0x0004) [0x0000000000000000] [0x00100000]
unsigned long          bCompleteOnHitGoal : 1;                  // 0x00D8 (0x0004)
[0x0000000000000000] [0x00200000]
unsigned long          bFailOnHitGoal : 1;                  // 0x00D8 (0x0004)
[0x0000000000000000] [0x00400000]
unsigned long          bFreezeCarOnSpawn : 1;                // 0x00D8 (0x0004)
[0x0000000000000000] [0x00800000]
unsigned long          bCanEverDodge : 1;                  // 0x00D8 (0x0004)
[0x0000000000000000] [0x01000000]
unsigned long          bCanEverDoubleJump : 1;            // 0x00D8 (0x0004)
[0x0000000000000000] [0x02000000]
unsigned long          bTutorialFinished : 1;              // 0x00D8 (0x0004)
[0x0000000000000000] [0x04000000]
unsigned long          bPauseDuringMessages : 1;            // 0x00D8 (0x0004)
[0x0000000000000000] [0x08000000]
unsigned long          bCanResetTutorial : 1;            // 0x00D8 (0x0004)
[0x0000000000000000] [0x10000000]
unsigned long          bCanSkipTutorial : 1;            // 0x00D8 (0x0004)
[0x0000000000000000] [0x20000000]
unsigned long          bShowGameFinishMessage : 1;        // 0x00D8
(0x0004) [0x0000000000000000] [0x40000000]
unsigned long          bSkipped : 1;                  // 0x00D8 (0x0004)
[0x0000000000000000] [0x80000000]
unsigned long          bQuit : 1;                  // 0x00DC (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bReset : 1;                  // 0x00DC (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long          bTLog : 1;                  // 0x00DC (0x0004)
[0x0000000000000000] [0x00000004]
struct FMessageInfo          CorrectAnswer;           // 0x00E0 (0x0040)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FMessageInfo          IncorrectAnswer;         // 0x0120 (0x0040)
[0x0000000000400000] (CPF_NeedCtorLink)
float                      StartFadeDisplayTime;       // 0x0160 (0x0004)
[0x0000000000000000]
struct FColor               FadeColor;             // 0x0164 (0x0004)
[0x0000000000000000]
float                      FadeTime;              // 0x0168 (0x0004)
```

```

[0x0000000000000000]
class UGFxShell_TA*           Shell;          // 0x0170 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FName                   ActionNameToDetect; // 0x0178 (0x0008)

float                          FailTimeAfterBallTouch; // 0x0180 (0x0004)
float                          CountdownTime;        // 0x0184 (0x0004)
int32_t                        BotTeam;          // 0x0188 (0x0004)
float                          TutorialFailTime; // 0x018C (0x0004)
float                          TutorialCompleteTime; // 0x0190 (0x0004)
float                          BallDestroyTimeOnTouch; // 0x0194 (0x0004)
float                          AllMessagesDisplayedTime; // 0x0198 (0x0004)
float                          WaitTimeBeforeCheckCompleteAction; // 0x019C (0x0004)
uint8_t                         TutorialStatus; // 0x01A0 (0x0001)
float                          StartTime;          // 0x01A4 (0x0004)
class FString                  StartTimeString; // 0x01A8 (0x0010)
TArray<class FString>         UseActions;        // 0x01B8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName>           IgnoreActionPressList; // 0x01C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName>           FailedActionPressList; // 0x01D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName>           StartActionPressList; // 0x01E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName>           CompleteActionPressList; // 0x01F8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FName>           CompleteActionIgnorePressList; // 0x0208
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __EventTutorialEnded__Delegate; // 0x0218
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate          __EventTutorialCompleted__Delegate; // 0x0230
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Tutorial_TA");
}
}
```

```
return uClassPointer;
};

class FString GetTutorialName();
void SkipTutorial_Confirm(class UGFxModal_X* Modal);
void SkipTutorial();
void ResetTutorial();
void SelectMultipleChoiceButton(class UGFxModal_X* Modal, int32_t buttonIndex);
void MultipleChoice_Button3(class UGFxModal_X* Modal);
void MultipleChoice_Button2(class UGFxModal_X* Modal);
void MultipleChoice_Button1(class UGFxModal_X* Modal);
void MultipleChoice_Button0(class UGFxModal_X* Modal);
void ForceActivateOutput(uint8_t Output);
void NotifyEventOnTutorial(class FString EventName);
void ReachedTimeAfterBallTouch();
void HandleBallDestroyed(class APawn_X* Pawn);
void HandleCarTouch(class ABall_TA* Ball, class ACar_TA* HitCar, uint8_t HitType);
void HandleHitGoal(class ABall_TA* Ball, class UGoal_TA* Goal);
void HandleBallAdded(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* Ball);
void RegisterGameBallDelegates();
void CloseMessageModal();
void TutorialCompleteTimeFinished();
void TutorialFailTimeFinished();
void EndTutorial();
void ShowGameMessage(struct FName Type, class FString Message);
void FailedTutorial();
void CompletedTutorial();
void SetIgnoreOnCompleteActionList();
void AddKeyNameFromAction(struct FName ActionName, TArray<class FString>& KeyList);
TArray<class FString> GetActionDisplayNames(struct FActionNameInfo ActionInfo);
bool RemoveMatchingKeys(class FString KeyToAdd, TArray<class FString>& Out_KeyList);
class FString GetUIReadableString(class FString Key);
class FString GetMergedActionString(TArray<class FString> KeyList);
class FString GetBodyText(struct FMessageInfo InMessage);
bool DisplayMessage(struct FMessageInfo InMessage);
bool DisplayNextMessage();
void DisplayCurrentMessage();
void OnAllMessagesDisplayed();
void ClearAndShowNextMessage(class UGFxModal_X* Modal);
void ShowNextMessage(class UGFxModal_X* Modal);
struct FMessageInfo GetMessage();
void eventTick(float DeltaTime);
void CheckActionNameToDetect();
void OnActionNameToDetectReleased();
void OnActionNameToDetectPressed();
void OnActionKeyReleased(struct FName ActionName);
void OnActionKeyPressed(struct FName ActionName);
bool NotifyKeyInput(int32_t ControllerId, struct FName Key, uint8_t EventType, float AmountDepressed, unsigned long bGamepad);
bool NotifyAxisInput(int32_t ControllerId, struct FName Key, float delta, float DeltaTime, unsigned long bGamepad);
bool CompleteActionPressed(struct FName Key, unsigned long bGamepad);
bool StartActionPressed(struct FName Key, unsigned long bGamepad);
void SetIgnoredInput(unsigned long bClearPreviousActions);
```

```

void SetAllDriving(uint8_t PlayerDriveType, unsigned long bDriving);
void TutorialStart();
void CountTimeElapsed();
void InitTutorial();
void OnScreenFadedOut();
void OnScreenFadedIn();
void HideFadeScreen();
void ShowFadeScreen(float DisplayTime);
void Destroyed();
bool CanDestroy();
void FoundBot();
void HandleVehicleSetup(class AGameEvent_Tutorial_TA* InGameEvent, class ACar_TA* Car);
void HandleInputTypeChanged(class UGfxShell_X* InShell);
void Init(class APlayerController_TA* InPC, class AGameEvent_Tutorial_TA* InGameEvent, class USequenceAction* InKNode_Owner);
void EventTutorialCompleted(class UTutorial_TA* Tutorial);
void EventTutorialEnded(class UTutorial_TA* Tutorial);
};

// Class TAGame.RPC_GetChallengeProgress_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_GetChallengeProgress_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    TArray<int32_t>               ChallengeIDs;     // 0x0130 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)
    TArray<struct FChallengeProgress> ProgressData;   // 0x0140
    (0x0010) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_GetChallengeProgress_TA");
        }

        return uClassPointer;
    }

    class URPC_GetChallengeProgress_TA* SetRequestedIDs(TArray<int32_t>& Ids);
    class URPC_GetChallengeProgress_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RocketPass_TA
// 0x0118 (0x0060 - 0x0178)
class URocketPass_TA : public UObject
{
public:
    TArray<struct FRocketPassRewardData>      FreeRewards;      // 0x0060

```

```

(0x0010) [0x0009000000400000] (CPF_NeedCtorLink)
TArray<struct FRocketPassRewardData> PremiumRewards; // 0x0070
(0x0010) [0x0009000000400000] (CPF_NeedCtorLink)
TArray<struct FRocketPassRewardData> PrestigeRewards; // 0x0080
(0x0010) [0x0009000000400000] (CPF_NeedCtorLink)
int32_t TierCap; // 0x0090 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t PrestigeTierCap; // 0x0094 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
unsigned long bIsPassActive : 1; // 0x0098 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long bOwnsPremium : 1; // 0x0098 (0x0004)
[0x0001000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long bPrestigeItemsDirtied : 1; // 0x0098 (0x0004)
[0x0001000000000000] [0x00000004]
float XPMultiplier; // 0x009C (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t TierLevel; // 0x00A0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t SecondsRemaining; // 0x00A4 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
class URocketPassBundleContainer_TA* Bundles; // 0x00A8
(0x0008) [0x0001004004080008] (CPF>ExportObject | CPF>Component | CPF>EditInline)
class UAsyncTask* SyncAllInfoTask; // 0x00B0 (0x0008)
[0x000100000002000] (CPF>Transient)
class UAsyncTask* PendingPlayerInfoTask; // 0x00B8 (0x0008)
[0x000100000002000] (CPF>Transient)
class UAsyncTask* PendingRewardsTask; // 0x00C0 (0x0008)
[0x000100000002000] (CPF>Transient)
class UAsyncTask* PendingPrestigeRewardsTask; // 0x00C8
(0x0008) [0x000100000002000] (CPF>Transient)
class UAsyncTask* PurchasePremiumTask; // 0x00D0 (0x0008)
[0x000100000002000] (CPF>Transient)
class UAsyncTask* PurchaseTiersTask; // 0x00D8 (0x0008)
[0x000100000002000] (CPF>Transient)
int32_t CodeRedemptionPurchasableID; // 0x00E0 (0x0004)
[0x00010000000002] (CPF>Const)
class URocketPassConfig_TA* RocketPassConfig; // 0x00E8
(0x0008) [0x0001800000002000] (CPF>Transient)
class UOnlineGame_X* OnlineGame; // 0x00F0 (0x0008)
[0x0001800000002000] (CPF>Transient)
class UWallet_TA* Wallet; // 0x00F8 (0x0008)
[0x0001000000002000] (CPF>Transient)
struct FScriptDelegate __EventError__Delegate; // 0x0100 (0x0018)
[0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate __EventRocketPassInfoChanged__Delegate; // 0x0118
(0x0018) [0x0000000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate __FreeRewards__ChangeNotify; // 0x0130
(0x0018) [0x0001000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate __PremiumRewards__ChangeNotify; // 0x0148
(0x0018) [0x0001000000400000] (CPF>NeedCtorLink)
struct FScriptDelegate __PrestigeRewards__ChangeNotify; // 0x0160
(0x0018) [0x0001000000400000] (CPF>NeedCtorLink)

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RocketPass_TA");
}

return uClassPointer;
};

void __RocketPass_TA__BeginState_0x1(class UWallet_TA* InWallet);
void __RocketPass_TA__RequestPlayerInfo_0x1();
void __RocketPass_TA__RequestRewards_0x1();
void __RocketPass_TA__RequestPrestigeRewards_0x1();
void __RocketPass_TA__PurchasePremium_0x2();
void __RocketPass_TA__PurchasePremium_0x1(class URPC_X* RPC);
void __RocketPass_TA__PurchaseTiers_0x2();
void __RocketPass_TA__PurchaseTiers_0x1(class URPC_X* RPC);
bool __RocketPass_TA__HandlePurchaseTiers_0x1(struct FRocketPassUnlock Unlock);
void __RocketPass_TA__CheckForPremiumProduct_0x1(class UPlayer* P);
bool __RocketPass_TA__GetPremiumProduct_0x1(class UOnlineProduct_TA* Op);
void __PrestigeRewards__ChangeNotifyFunc();
void __PremiumRewards__ChangeNotifyFunc();
void __FreeRewards__ChangeNotifyFunc();
class UOnlineProduct_TA* GetPremiumProduct(TArray<class UOnlineProduct_TA*>& OnlineProducts);
void HandleRedeemedOnlineProducts(class USaveData_TA* Data, TArray<class UOnlineProduct_TA*>& RedeemedOnlineProducts);
void CheckForPremiumProduct();
bool HasFreeRewards();
bool HasPlayerInfo();
TArray<class UAsyncTask*> RetrieveMissingData();
void HandleReceivedRewardDrop(class URewardDrop_XPMultiplier_TA* InXPMultiplier);
void AddPrestigeUnlock(struct FRocketPassUnlock Unlock);
void GiveUnlocks(TArray<struct FRocketPassUnlock>& Unlocks);
void HandlePurchaseTiers(class URPC_RocketPassPurchaseTiers_TA* RPC);
class UAsyncTask* PurchaseTiers(int32_t PurchasableID, TArray<struct FProductInstanceID>& KeyInstanceIDs);
void HandlePurchasePremium(class URPC_RocketPassPurchasePremium_TA* RPC);
class UAsyncTask* PurchasePremium(int32_t PurchasableID, TArray<struct FProductInstanceID>& KeyInstanceIDs);
void HandleGetPrestigeRewards(class URPC_RocketPassGetPlayerPrestigeRewards_TA* RPC);
class UAsyncTask* RequestPrestigeRewards();
void HandleGetRewards(class URPC_RocketPassGetRewardContent_TA* RPC);
class UAsyncTask* RequestRewards();
void ForceSetRocketPassInfo(struct FRocketPassInfo Info, struct FRocketPassStore Store);
void SetRocketPassInfo(struct FRocketPassInfo Info);
void HandleGetPlayerInfo(class URPC_RocketPassGetPlayerInfo_TA* RPC);
class UAsyncTask* RequestPlayerInfo();
void UpdateSecondsRemaining();
void RequestAllInfo();
```

```

void UpdateState();
void HandleEventConfigChanged();
void HandlePsyNetLogout();
void HandlePsyNetLogin();
void eventConstruct();
void EventRocketPassInfoChanged(class URocketPass_TA* RocketPass);
void EventError(class UError* InError);
};

// Class TAGame.RPC_Challenge_ClearNewInfo_TA
// 0x0058 (0x00E8 - 0x0140)
class URPC_Challenge_ClearNewInfo_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<int32_t>               ChallengeIDs;      // 0x0130 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_ClearNewInfo_TA");
        }

        return uClassPointer;
    }

    class URPC_Challenge_ClearNewInfo_TA* SetChallengeIDs(TArray<int32_t> InChallengeIDs);
    class URPC_Challenge_ClearNewInfo_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_Challenge_AvailableReceived_TA
// 0x004C (0x00E8 - 0x0134)
class URPC_Challenge_AvailableReceived_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)
    int32_t                      Id;                 // 0x0130 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_AvailableReceived_TA");
        }

        return uClassPointer;
    }
}

```

```

}

return uClassPointer;
};

class URPC_Challenge_AvailableReceived_TA* SetChallengeID(int32_t ChallengeID);
class URPC_Challenge_AvailableReceived_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_Challenge_CompleteReceived_TA
// 0x004C (0x00E8 - 0x0134)
class URPC_Challenge_CompleteReceived_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x000000000000400000] (CPF_NeedCtorLink)
int32_t                         Id;                // 0x0130 (0x0004)
[0x000000000000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_CompleteReceived_TA");
}

return uClassPointer;
};

class URPC_Challenge_CompleteReceived_TA* SetChallengeID(int32_t ChallengeID);
class URPC_Challenge_CompleteReceived_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.GameEvent_Tutorial_Basic_TA
// 0x000C (0x0EE0 - 0x0EEC)
class AGameEvent_Tutorial_Basic_TA : public AGameEvent_Tutorial_TA
{
public:
int32_t           CurrentScore;           // 0x0EE0 (0x0004)
[0x0000000000000000]
int32_t           TotalRounds;           // 0x0EE4 (0x0004)
[0x0000000000000000]
int32_t           CurrentRounds;          // 0x0EE8 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Tutorial_Basic_TA");
}

return uClassPointer;
};

bool MoveToGround(class AActor* Mover, float HeightCheck);
void OnVehicleSetup(class ACar_TA* Car);
void InitBallVelocity();
void CheckForReset();
void StartRound();
void HideTimer();
void SetManualScores(int32_t InTotalRounds, int32_t InCurrentRound, int32_t InCurrentScore);
int32_t GetGameEventRounds();
int32_t GetTotalRounds();
int32_t GetScore();
void SkipTutorial();
class ATeam_TA* PickTeam(class AController* C);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct
FRotator& out_Rotation);
};

// Class TAGame.GameEvent_Tutorial_Advanced_TA
// 0x0004 (0x0EEC - 0x0EF0)
class AGameEvent_Tutorial_Advanced_TA : public AGameEvent_Tutorial_Basic_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Tutorial_Advanced_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_Challenge_TutorialComplete_TA
// 0x0058 (0x00E8 - 0x0140)
class URPC_Challenge_TutorialComplete_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 TutorialType;       // 0x0130 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_TutorialComplete_TA");
}

return uClassPointer;
};

class URPC_Challenge_TutorialComplete_TA* SetTutorialType(class FString Type);
class URPC_Challenge_TutorialComplete_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_Challenge_TrainingComplete_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_Challenge_TrainingComplete_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 TrainingType;        // 0x0130 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 Difficulty;         // 0x0140 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t                       Score;              // 0x0150 (0x0004)
[0x0000000000000000]
int32_t                       TotalRounds;       // 0x0154 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_TrainingComplete_TA");
}

return uClassPointer;
};

class URPC_Challenge_TrainingComplete_TA* SetFromTraining(class AGameEvent_Training_TA* Training);
class URPC_Challenge_TrainingComplete_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.ChampionshipTrophy_TA
// 0x0010 (0x0060 - 0x0070)
class UChampionshipTrophy_TA : public UObject

```

```

{
public:
class AScreenDisplayActor_TA*           ChampionshipTrophyArchetype;      // 0x0060
(0x0008) [0x0000000000000000]
class UAkSoundCue*                     ChampionshipTrophySound;          // 0x0068
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChampionshipTrophy_TA");
}

return uClassPointer;
};

void Spawn(class AActor* Owner);

// Class TAGame.ScreenDisplayActor_TA
// 0x0064 (0x02C8 - 0x032C)
class AScreenDisplayActor_TA : public ADynamicSMActor_Spawnable
{
public:
class APlayerControllerBase_TA*          PC;                                // 0x02C8 (0x0008)
[0x0000000000000000]
struct FVector                         StartLocationOffset;            // 0x02D0 (0x000C)
[0x0000000000000000]
struct FVector                         LocationOffset;                // 0x02DC (0x000C)
[0x0000000000000000]
struct FRotator                        CurrentRotation;              // 0x02E8 (0x000C)
[0x0000000000000000]
float                                 RotationSpeed;                 // 0x02F4 (0x0004)
[0x0000000000000000]
float                                 StartRotationOffset;           // 0x02F8 (0x0004)
[0x0000000000000000]
float                                 BlendInSpeed;                 // 0x02FC (0x0004)
[0x0000000000000000]
float                                 BlendInAmount;                // 0x0300 (0x0004)
[0x0000000000000000]
float                                 CameraShakeDuration;          // 0x0304 (0x0004)
[0x0000000000000000]
float                                 CameraShakeAmplitude;         // 0x0308 (0x0004)
[0x0000000000000000]
float                                 CameraShakeFrequency;          // 0x030C (0x0004)
[0x0000000000000000]
class UParticleSystem*                  ParticleFX;                  // 0x0310 (0x0008)
[0x0000000000000000]
class UParticleSystemComponent*        ParticleComponent;             // 0x0318
(0x0008) [0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

```

```
struct FVector           FXOffset;          // 0x0320 (0x000C)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ScreenDisplayActor_TA");
}

return uClassPointer;
};

void SpawnFX();
void PlayBlendInCameraShake();
void OnOffsetBlendedIn();
void Tick(float DeltaTime);
void Init();
};

// Class TAGame.ChatConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UChatConfig_TA : public UOnlineConfig_X
{
public:
TArray<int32_t>           MaxMessagesPerChannel;      // 0x0078 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ChatConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.StoredChatData_TA
// 0x006C (0x0060 - 0x00CC)
class UStoredChatData_TA : public UObject
{
public:
uint8_t                    ChatChannel;           // 0x0060 (0x0001)
[0x0000000000000000]
struct FUniqueNetId        SenderId;            // 0x0068 (0x0048)
```

```
[0x0000000000400000] (CPF_NeedCtorLink)
uint64_t LastChatUpdate; // 0x00B0 (0x0008)
[0x0000000000000000]
TArray<struct FGFxChatMessage> Messages; // 0x00B8
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
unsigned long bIsActive : 1; // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StoredChatData_TA");
}

return uClassPointer;
};

void RepopulateChat(class UGFxData_Chat_TA* GFxChat);
void StoreChatMessage(struct FGFxChatMessage& Data);
};

// Class TAGame.CheatManagerBase_TA
// 0x0018 (0x0098 - 0x00B0)
class UCheatManagerBase_TA : public UCheatManager_X
{
public:
int32_t LastEquippedProductID; // 0x0098 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FProductInstanceId LastEquippedInstanceId; // 0x00A0
(0x0010) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CheatManagerBase_TA");
}

return uClassPointer;
};

};

// Class TAGame.CheatManager_TA
// 0x0020 (0x00B0 - 0x00D0)
class UCheatManager_TA : public UCheatManagerBase_TA
{
```

```
public:
class AInputRecorder_TA*           InputRecorder;          // 0x00B0 (0x0008)
[0x0000000000000000] (CPF_Transient)
struct FColor                      DebugColor;           // 0x00B8 (0x0004)
[0x0000000000000000] (CPF_Transient)
int32_t                           TestClubColor;        // 0x00BC (0x0004)
[0x0000000000000000] (CPF_Transient)
float                             BodyHandlingTestTime; // 0x00C0 (0x0004)
[0x0000000000000000]
int32_t                           BodyHandlingTestIndex; // 0x00C4 (0x0004)
[0x0000000000000000]
class UTickNotifyDelegate_TA*     TickNotifier;         // 0x00C8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CheatManager_TA");
}

return uClassPointer;
};

};

// Class TAGame.CheckoutErrors_TA
// 0x0028 (0x0080 - 0x00A8)
class UCheckoutErrors_TA : public UErrorList
{
public:
class UErrorType*                MtxConfigDisabled;    // 0x0080 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*                SystemOverlayDisabled; // 0x0088 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*                TransactionInProgress; // 0x0090 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*                InvalidNumberOfItemsInCart; // 0x0098 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*                InvalidItemsInCart;      // 0x00A0 (0x0008)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CheckoutErrors_TA");
}
}
```

```
return uClassPointer;
};

};

// Class TAGame.GFxModal_Cinematic_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Cinematic_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Cinematic_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_CinematicIntroSeqFinished_TA
// 0x0008 (0x017C - 0x0184)
class USeqEvent_CinematicIntroSeqFinished_TA : public USequenceEvent
{
public:
int32_t SequenceIndex; // 0x0180 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_CinematicIntroSeqFinished_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_CinematicIntro_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_CinematicIntro_TA : public UGFxDataSingleton_X
{
```

```
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_CinematicIntro_TA");
}

return uClassPointer;
};

void __GFxData_CinematicIntro_TA__OnShellSet_0x1(class
USeqAct_CinematicIntroStartNextSeq_TA* SeqAct);
class UCinematicIntroSequence_TA* GetActiveSequence();
void HandleSequenceFinished(class UCinematicIntroSequence_TA* Seq);
void OnFinished();
void StartSequence(int32_t SequenceIndex);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.CinematicsCar_TA
// 0x0000 (0xB4860 - 0xB4860)
class ACinematicsCar_TA : public ACar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CinematicsCar_TA");
}

return uClassPointer;
};

void HandleMatineeAssetLoaded(class UProductLoader_TA* Loader, class UProductAsset_TA*
Asset);
void SetupLoadout(class ASkeletalMeshActorMAT_Products_TA* SkelMeshActor);
};

// Class TAGame.ClientConnectionTracker_TA
// 0x006C (0x0A4 - 0x110)
class UClientConnectionTracker_TA : public UActorComponent_X
{
public:
```

```

unsigned long          bDisconnectedFromServer : 1;           // 0x00A8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bShowConnectionQualityIndicators : 1;    // 0x00A8
(0x0004) [0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
uint8_t                LatencyStatus;                      // 0x00AC (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t                JitterStatus;                       // 0x00AD (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t                PacketStatus;                      // 0x00AE (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
float                 LatencyLastUpdateTime;             // 0x00B0 (0x0004)
[0x0000000000000000]
float                 JitterRollingAverage;            // 0x00B4 (0x0004)
float                 JitterLastUpdateTime;              // 0x00B8 (0x0004)
float                 PacketLastUpdateTime;             // 0x00BC (0x0004)
float                 NewPacketsRollingAverage;        // 0x00C0 (0x0004)
float                 PacketsLostRollingAverage;        // 0x00C4 (0x0004)
float                 RollingPercentLost;               // 0x00C8 (0x0004)
float                 PriorTotalPacketsInOut;          // 0x00CC (0x0004)
float                 PriorTotalPacketsLost;            // 0x00D0 (0x0004)
float                 RollingBadPingTime;              // 0x00D4 (0x0004)
float                 RollingBadJitterTime;            // 0x00D8 (0x0004)
float                 RollingBadPacketLossTime;         // 0x00DC (0x0004)
[0x0000004000000000]
class UConnectionQualityConfig_TA*      ConnectionQualityConfig; // 0x00E0
(0x0008) [0x00080000000001] (CPF>Edit)
class APRI_TA*                     TrackedPRI;           // 0x00E8 (0x0008)
[0x0000000000000000]
class UConnectionQualitySave_TA*       ConnectionQualitySave; // 0x00F0
(0x0008) [0x0000000000000000]
struct FScriptDelegate             __EventNetStatsPolled__Delegate; // 0x00F8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClientConnectionTracker_TA");
}

return uClassPointer;

```

```

};

uint8_t GetWorstConnectionStatusBeyondLatency();
static uint8_t GetConnectionStatus(float InMetricValue, float MediocreBoundary, float
BadBoundary);
static bool ShouldUpdateMetric(uint8_t ExistingValue, uint8_t NewValue, float LastUpdateTime,
float CurrentTime, float DelayTime);
static void PollMetric(uint8_t NewState, float NewTime, float DelayTime, uint8_t& ExistingState,
float& UpdateTime);
void UpdateSavedRollingMetricTimes();
void UpdateRollingConnectionQuality(uint8_t Status, float& RollingTime);
void UpdateRollingConnectionQualities();
float ComputeRollingPacketLossPercent();
float ComputeRollingJitter();
void PollConnectionStats();
void HandleConnectionQualitySaveUnloaded(class UConnectionQualitySave_TA*
InConnectionQualitySave);
void HandleConnectionQualitySaveLoaded(class UConnectionQualitySave_TA*
InConnectionQualitySave);
void SubscribeToConnectionQualitySave(class USaveData_TA* SaveData);
void ResetConnectionStatuses();
void StopTrackingConnection();
void BeginTrackingConnection();
void SetTrackedPRI(class APRI_TA* InPRI);
void eventDetached();
void eventAttached();
void EventNetStatsPolled(class UClientConnectionTracker_TA* InTracker, struct
FClientConnectionStats InStats);
};

// Class TAGame.ConnectionQualitySave_TA
// 0x001C (0x00C8 - 0x00E4)
class UConnectionQualitySave_TA : public USaveObject_TA
{
public:
float [0x0000000000000000] RollingBadPingTime; // 0x00C8 (0x0004)
float [0x0000000000000000] RollingBadJitterTime; // 0x00CC (0x0004)
float [0x0000000000000000] RollingBadPacketLossTime; // 0x00D0 (0x0004)
uint64_t [0x0000000000000000] LastNotificationEpochSeconds; // 0x00D8 (0x0008)
unsigned long bShowConnectionQualityIndicators : 1; // 0x00E0
(0x0004) [0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ConnectionQualitySave_TA");
}
}

```

```

}

return uClassPointer;
};

};

// Class TAGame.ConnectionQualityConfig_TA
// 0x0060 (0x0078 - 0x00D8)
class UConnectionQualityConfig_TA : public UOnlineConfig_X
{
public:
float [0x0000000000000000] ClientPollRate; // 0x0078 (0x0004)
float [0x0000000000000000] ClientDelayBeforePolling; // 0x007C (0x0004)
float [0x0000000000000000] ClientUpdateDelay; // 0x0080 (0x0004)
float [0x0000000000000000] MediocrePing; // 0x0084 (0x0004)
float [0x0000000000000000] BadPing; // 0x0088 (0x0004)
float [0x0000000000000000] RecentBadPingFlagPercent; // 0x008C (0x0004)
float [0x0000000000000000] BadAckTime; // 0x0090 (0x0004)
float [0x0000000000000000] BadReceiveTime; // 0x0094 (0x0004)
float [0x0000000000000000] MediocreJitter; // 0x0098 (0x0004)
float [0x0000000000000000] BadJitter; // 0x009C (0x0004)
float [0x0000000000000000] JitterRollingAverageTimespan; // 0x00A0 (0x0004)
float [0x0000000000000000] RecentBadJitterFlagPercent; // 0x00A4 (0x0004)
float [0x0000000000000000] MediocrePacketLossPercent; // 0x00A8 (0x0004)
float [0x0000000000000000] BadPacketLossPercent; // 0x00AC (0x0004)
float [0x0000000000000000] PacketLossRollingAverageTimespan; // 0x00B0 (0x0004)
float [0x0000000000000000] RecentBadPacketLossFlagPercent; // 0x00B4 (0x0004)
float [0x0000000000000000] RecentPoorConnectionRollingTimespan; // 0x00B8 (0x0004)
float [0x0000000000000000] NotificationDelayInDays; // 0x00BC (0x0004)

unsigned long bJoinGameConnectionCheckEnabled : 1; // 0x00C0
(0x0004) [0x0000000000000000] [0x00000001]
unsigned long bShouldProvidePoorConnectionHelp : 1; // 0x00C0
(0x0004) [0x0000000000000000] [0x00000002]
float BadPRIReadyTime; // 0x00C4 (0x0004)
[0x0000000000000000]

```

```
class FString           PoorConnectionHelpNotificationURL;      // 0x00C8
(0x0010) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ConnectionQualityConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.ClientJitterBuffer_STS_TA
// 0x0008 (0x007C - 0x0084)
class UClientJitterBuffer_STS_TA : public UClientJitterBuffer_TA
{
public:
float          PrevBufferTime;          // 0x0080 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClientJitterBuffer_STS_TA");
}

return uClassPointer;
};

void ReceivedPacket();
};

// Class TAGame.ClientStayAsPartyVoteBeginEvent_TA
// 0x0000 (0x0060 - 0x0060)
class UClientStayAsPartyVoteBeginEvent_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
```

```

{
uClassPointer = UObject::FindClass("Class TAGame.ClientStayAsPartyVoteBeginEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.XPManager_TA
// 0x0020 (0x0060 - 0x0080)
class UXPManger_TA : public UObject
{
public:
int32_t MultiplierCapLevel; // 0x0060 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
int32_t XPPerLevelMultiplier; // 0x0064 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
int32_t XPPerLevelCap; // 0x0068 (0x0004)
[0x0000000000000003] (CPF_Edit | CPF_Const)
TArray<struct FTitleInfo> Titles; // 0x0070 (0x0010)
[0x000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.XPManager_TA");
}

return uClassPointer;
};

static int32_t CalculateTotalXPRequiredForLevel(int32_t Level);
class FString DetermineLevelTitle(int32_t Level);
static int32_t CalculateRequiredXPForLevel(int32_t Level);
static int32_t CalculatePlayerLevel(int32_t XP, unsigned long bShowLog);
};

// Class TAGame.RewardDrop_TA
// 0x0008 (0x0060 - 0x0068)
class URewardDrop_TA : public UObject
{
public:
struct FName RewardName; // 0x0060 (0x0008)
[0x0000000040000002] (CPF_Const | CPF_EditInLineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RewardDrop_TA");
}

return uClassPointer;
};

struct FOnlineXPReward OnlineXPReward();
};

// Class TAGame.RewardDrop_XP_TA
// 0x0098 (0x0068 - 0x0100)
class URewardDrop_XP_TA : public URewardDrop_TA
{
public:
float Total; // 0x0068 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t Base; // 0x006C (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t BaseLevel; // 0x0070 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t Level; // 0x0074 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
TArray<struct FGFxLevelThreshold> LevelThresholds; // 0x0078
(0x0010) [0x0000000040400000] (CPF>NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FOnlineXPMODIFIER> Modifiers; // 0x0088 (0x0010)
[0x0000000040400000] (CPF>NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FName> PremiumModifierNames; // 0x0098
(0x0010) [0x0000000000400003] (CPF>Edit | CPF>Const | CPF>NeedCtorLink)
TArray<struct FName> PenaltyModifierNames; // 0x00A8
(0x0010) [0x0000000000400003] (CPF>Edit | CPF>Const | CPF>NeedCtorLink)
TArray<struct FName> PromoModifierNames; // 0x00B8
(0x0010) [0x0000000000400003] (CPF>Edit | CPF>Const | CPF>NeedCtorLink)
struct FName WeeklyCapReached; // 0x00C8 (0x0008)
[0x0000000000000002] (CPF>Const)
class FString RewardModifiers_SectionName; // 0x00D0 (0x0010)
[0x0000000000400002] (CPF>Const | CPF>NeedCtorLink)
class FString RewardModifiers_PackageName; // 0x00E0
(0x0010) [0x0000000000400002] (CPF>Const | CPF>NeedCtorLink)
struct FName WeeklyCap; // 0x00F0 (0x0008)
[0x0000000000000002] (CPF>Const)
struct FName RestXP; // 0x00F8 (0x0008)
[0x0000000000000002] (CPF>Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RewardDrop_XP_TA");
}
}

```

```

}

return uClassPointer;
};

static struct FGFxLevelThreshold __RewardDrop_XP_TA__CreateInstance_0x1(int32_t XP);
int32_t __RewardDrop_XP_TA__OnlineXPReward_0x1(struct FGFxLevelThreshold XP);
struct FOnlineXPReward OnlineXPReward();
static class FString LocalizeModifier(class URewardDrop_XP_TA* RewardDrop, struct
FOnlineXPMODIFIER Modifier);
static class URewardDrop_TA* CreateInstance(struct FOnlineXPReward& XPDrop);
};

// Class TAGame.GFxData_ClubInviteNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ClubInviteNotification_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClubInviteNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.ClubNotificationsSync_TA
// 0x0000 (0x0060 - 0x0060)
class UClubNotificationsSync_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClubNotificationsSync_TA");
}

return uClassPointer;
};

void HandleClubChanged(class UPsyNetClientService_X* Service);

```

```
void HandleClubUpdated(class UPsyNetService_ClubsUpdate_TA* Update);
void eventConstruct();
};

// Class TAGame.PsyNetService_ClubsUpdate_TA
// 0x0008 (0x0090 - 0x0098)
class UPsyNetService_ClubsUpdate_TA : public UPsyNetClientService_X
{
public:
class UClubDetails_X* ClubDetails; // 0x0090 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ClubsUpdate_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_ClubsOwnerChanged_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_ClubsOwnerChanged_TA : public UPsyNetClientService_X
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ClubsOwnerChanged_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_ClubsMemberRemoved_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_ClubsMemberRemoved_TA : public UPsyNetClientService_X
{
public:
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ClubsMemberRemoved_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_ClubsMemberAdded_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_ClubsMemberAdded_TA : public UPsyNetClientService_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ClubsMemberAdded_TA");
}

return uClassPointer;
};

};

// Class TAGame.ClubsConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UClubsConfig_TA : public UOnlineConfig_X
{
public:
int32_t TagLengthMin; // 0x0078 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t TagLengthMax; // 0x007C (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t NameLengthMin; // 0x0080 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t NameLengthMax; // 0x0084 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClubsConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.ClubUtil_TA
// 0x0000 (0x0060 - 0x0060)
class UClubUtil_TA : public UClubUtil_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ClubUtil_TA");
}

return uClassPointer;
};

static int32_t AccentColorToUIColorID(int32_t InColor);
static int32_t ColorToUIColorID(int32_t InColor);
static int32_t UIAccentColorIDToColor(int32_t ColorID);
static int32_t UIColorIDToColor(int32_t ColorID);
static struct FClubColorSet GetClubColorSet(class UClubDetails_X* Club);
static uint8_t EnsureDifferentColorsTA(struct FClubColorSet& Set0, struct FClubColorSet& Set1);
};

// Class TAGame.CollisionTestTrack_TA
// 0x0008 (0x0268 - 0x0270)
class ACollisionTestTrack_TA : public AActor
{
public:
class UCollisionTestComponent_TA* CollisionTestComponent; // 0x0268
(0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.CollisionTestTrack_TA");
}

return uClassPointer;
};

};

// Class TAGame.Constraint2D_TA
// 0x0000 (0x02A0 - 0x02A0)
class AConstraint2D_TA : public ARB_ConstraintActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Constraint2D_TA");
}

return uClassPointer;
};

};

// Class TAGame.ContentSave_TA
// 0x0018 (0x00C8 - 0x00E0)
class UContentSave_TA : public USaveObject_TA
{
public:
class UContentConfig_TA*           Config;           // 0x00C8 (0x0008)
[0x0000800000002000] (CPF_Transient)
TArray<struct FContentPair>        ContentMap2;      // 0x00D0 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ContentSave_TA");
}

return uClassPointer;
};

void HandleConfigChange();
void OnLoad();

```

```
};

// Class TAGame.ConvertPlatformFriendsEvent_TA
// 0x0000 (0x0060 - 0x0060)
class UConvertPlatformFriendsEvent_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ConvertPlatformFriendsEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.CountdownObject_TA
// 0x0024 (0x008C - 0x00B0)
class UCountdownObject_TA : public UTickableStateObject_X
{
public:
int32_t           CountTime;           // 0x0090 (0x0004)
[0x0000004000000001] (CPF_Edit)
struct FScriptDelegate   _EventCountChange__Delegate;    // 0x0098
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CountdownObject_TA");
}

return uClassPointer;
};

void UpdateCountdown();
void StopCount();
void Start(int32_t InCountTime);
void EventCountChange(class UCountdownObject_TA* CountdownObject, int32_t CountValue);
};

// Class TAGame.CrewDropGroup_TA
// 0x0008 (0x00C0 - 0x00C8)
```

```
class UCrewDropGroup_TA : public UItemDropGroup_TA
{
public:
    class UTexture*           Icon;                      // 0x00C0 (0x0008)
    [0x0000000040000000] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CrewDropGroup_TA");
        }

        return uClassPointer;
    }

};

// Class TAGame.GFxData_CrewDropGroup_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_CrewDropGroup_TA : public UGFxData_ItemDropGroup_TA
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_CrewDropGroup_TA");
        }

        return uClassPointer;
    }

};

// Class TAGame.CrowdManager_TA
// 0x0048 (0x0268 - 0x02B0)
class ACrowdManager_TA : public AActor
{
public:
    TArray<class ACrowdActor_TA*>      CrowdActors;          // 0x0268
    (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class UCrowdActorSettings_TA*       Settings;           // 0x0278 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    unsigned long                     bInitialized : 1;      // 0x0280 (0x0004)
    [0x0000000000002000] [0x00000001] (CPF_Transient)
    TArray<class ACrowdActor_TA*>      RandomChantCrowdActors; // 0x0288
```

```
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UAkSoundCue* ReplicatedGlobalOneShotSound; // 0x0298
(0x0008) [0x0000000100002020] (CPF_Net | CPF_Transient)
float DefenseZoneTime; // 0x02A0 (0x0004)
[0x0000000000002000] (CPF_Transient)
class AGameEvent_Soccar_TA* GameEvent; // 0x02A8
(0x0008) [0x0000000000002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdManager_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
void TimerUpdateDefenseChant();
void StartTimerDefenseChant();
void PlayRandomChant();
void SetRandomChantTimer();
void PlayMatchEndedEncore();
void HandleMatchEnded(class AGameEvent_Soccar_TA* G);
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleGameStateChanged(class AGameEvent_TA* G);
void ClearGlobalOneShotSound();
void PlayGlobalOneShotSound(class UAkSoundCue* Sound);
void Init(class AGameEvent_Soccar_TA* InGameEvent);
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.CrowdSoundManager_Knockout_TA
// 0x0008 (0x0138 - 0x0140)
class UCrowdSoundManager_Knockout_TA : public UCrowdSoundManagerBase_TA
{
public:
class UCrowdSoundSettings_Knockout_TA* Settings; // 0x0138
(0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundManager_Knockout_TA");
```

```

}

return uClassPointer;
};

void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0xa(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x9(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x8(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x7(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x6(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x5(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x4(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x3(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x2(class
ACrowdActor_TA* C);
void __CrowdSoundManager_Knockout_TA__HandlePlayerStatEvent_0x1(class
ACrowdActor_TA* C);
class UCrowdSoundSettingsBase_TA* GetBaseCrowSettings();
void HandlePlayerStatEvent(class APRI_TA* InPRI, class UStatEvent_TA* StatEvent, int32_t
Count);
void eventDetached();
void eventAttached();
};

// Class TAGame.CrowdSoundSettings_Knockout_TA
// 0x0050 (0x0100 - 0x0150)
class UCrowdSoundSettings_Knockout_TA : public UCrowdSoundSettingsBase_TA
{
public:
class UDYNAMICValueModifierDuration_X*      ExcitemenHeavyHit;           // 0x0100
(0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICValueModifierDuration_X*      ExcitemenLightHit;          // 0x0108
(0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICValueModifierDuration_X*      ExcitemenAerialHeavyHit;     //
0x0110 (0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICValueModifierDuration_X*      ExcitemenAerialLightHit;     // 0x0118
(0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICValueModifierDuration_X*      ExcitemenHeavyBlock;         // 0x0120
(0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
CPF_EditInline)
class UDYNAMICValueModifierDuration_X*      ExcitemenLightBlock;         // 0x0128
(0x0008) [0x000100004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |

```

```

CPF>EditInline)
class UDynamicValueModifierDuration_X*      ExcitementKnockout;           // 0x0130
(0x0008) [0x000100004400009] (CPF>Edit | CPF>ExportObject | CPF>NeedCtorLink | 
CPF>EditInline)
class UDynamicValueModifierDuration_X*      ExcitementDeath;            // 0x0138
(0x0008) [0x000100004400009] (CPF>Edit | CPF>ExportObject | CPF>NeedCtorLink | 
CPF>EditInline)
class UDynamicValueModifierDuration_X*      ExcitementThrow;           // 0x0140
(0x0008) [0x000100004400009] (CPF>Edit | CPF>ExportObject | CPF>NeedCtorLink | 
CPF>EditInline)
class UDynamicValueModifierDuration_X*      ExcitementGrab;            // 0x0148
(0x0008) [0x000100004400009] (CPF>Edit | CPF>ExportObject | CPF>NeedCtorLink | 
CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundSettings_Knockout_TA");
}

return uClassPointer;
};

};

// Class TAGame.CrowdSounds_Knockout_TA
// 0x0194 (0x00FC - 0x0290)
class UCrowdSounds_Knockout_TA : public UCrowdSoundsBase_TA
{
public:
struct FCrowdExcitementSounds      HeavyHit;                // 0x0100 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      LightHit;               // 0x0128 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      AerialHeavyHit;        // 0x0150
(0x0028) [0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      AerialLightHit;         // 0x0178 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      HeavyBlock;             // 0x01A0
(0x0028) [0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      LightBlock;             // 0x01C8 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      KnockOut;              // 0x01F0 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      Death;                 // 0x0218 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      Throw;                 // 0x0240 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)
struct FCrowdExcitementSounds      Grab;                 // 0x0268 (0x0028)
[0x000100000400001] (CPF>Edit | CPF>NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSounds_Knockout_TA");
}

return uClassPointer;
};

};

// Class TAGame.CrowdSoundManager_TA
// 0x005C (0x0138 - 0x0194)
class UCrowdSoundManager_TA : public UCrowdSoundManagerBase_TA
{
public:
class UCrowdSoundSettings_TA*           Settings;          // 0x0138 (0x0008)
[0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
float             DefenseZoneTime;    // 0x0140 (0x0004)
[0x0001000000002000] (CPF_Transient)
float             LastBallHitNoiseTime; // 0x0144 (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t           LastBallHitNoiseLevel; // 0x0148 (0x0004)
[0x0001000000002000] (CPF_Transient)
float             TimeSinceLastGoal;   // 0x014C (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t           LeadChanges;       // 0x0150 (0x0004)
[0x0001000000002000] (CPF_Transient)
float             OpenGoalTime;       // 0x0154 (0x0004)
[0x0001000000002000] (CPF_Transient)
float             NearGoalTime;      // 0x0158 (0x0004)
[0x0001000000002000] (CPF_Transient)
float             FastBreakTime;     // 0x015C (0x0004)
[0x0001000000002000] (CPF_Transient)
unsigned long      bFastBreakInProgress : 1; // 0x0160 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)
unsigned long      bShotInProgress : 1;  // 0x0160 (0x0004)
[0x0001000000002000] [0x00000002] (CPF_Transient)
unsigned long      bAerialInProgress : 1; // 0x0160 (0x0004)
[0x0001000000002000] [0x00000004] (CPF_Transient)
unsigned long      bWaitingForKickoffTouch : 1; // 0x0160 (0x0004)
[0x0001000000002000] [0x00000008] (CPF_Transient)
unsigned long      bWaitingForTenseClear : 1; // 0x0160 (0x0004)
[0x0001000000002000] [0x00000010] (CPF_Transient)
class UGoal_TA*    ShotGoal;         // 0x0168 (0x0008)
[0x0001000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
float             ShotMaxScorability; // 0x0170 (0x0004)
[0x0001000000002000] (CPF_Transient)
float             LastShotTime;     // 0x0174 (0x0004)

```

```
[0x0001000000002000] (CPF_Transient)
float AerialDuration; // 0x0178 (0x0004)
[0x0001000000002000] (CPF_Transient)
float LastAerialTime; // 0x017C (0x0004)
[0x0001000000002000] (CPF_Transient)
float TenseNearGoalTime; // 0x0180 (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t LastScoreSeparation; // 0x0184 (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t ScoreSeparationTrend; // 0x0188 (0x0004)
struct FName LastScoreState; // 0x018C (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundManager_TA");
}

return uClassPointer;
};

float __CrowdSoundManager_TA__InitModifiers_0xe();
float __CrowdSoundManager_TA__InitModifiers_0xd();
float __CrowdSoundManager_TA__InitModifiers_0xc();
float __CrowdSoundManager_TA__InitModifiers_0xb();
float __CrowdSoundManager_TA__InitModifiers_0xa();
float __CrowdSoundManager_TA__InitModifiers_0x9();
float __CrowdSoundManager_TA__InitModifiers_0x8();
float __CrowdSoundManager_TA__InitModifiers_0x7();
float __CrowdSoundManager_TA__InitModifiers_0x6();
float __CrowdSoundManager_TA__InitModifiers_0x5();
float __CrowdSoundManager_TA__InitModifiers_0x4();
float __CrowdSoundManager_TA__InitModifiers_0x3();
float __CrowdSoundManager_TA__InitModifiers_0x2();
float __CrowdSoundManager_TA__InitModifiers_0x1();
class UAkSoundCue* __CrowdSoundManager_TA__GoalScored_0x3(class
UCrowdSoundsBase_TA* S);
void __CrowdSoundManager_TA__GoalScored_0x2(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__GoalScored_0x1(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__HandleDemolish_0x2(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__HandleDemolish_0x1(class ACrowdActor_TA* C);
float __CrowdSoundManager_TA__HandleGameTimeUpdated_0x1();
void __CrowdSoundManager_TA__HandleCarHitBall_0x2(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__HandleCarHitBall_0x1(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__CheckGoals_0x5(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__CheckGoals_0x4(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__CheckGoals_0x3(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__CheckGoals_0x2(class ACrowdActor_TA* C);
```

```

void __CrowdSoundManager_TA__CheckGoals_0x1(class ACrowdActor_TA* C);
void __CrowdSoundManager_TA__CheckAerial_0x1(class ACrowdActor_TA* C);
void PrintDebugInfo(class UDebugDrawer* Drawer);
class UCrowdSoundSettingsBase_TA* GetBaseCrowdSettings();
void CheckAerial(float DeltaTime);
void ClearShotInProgress();
void CheckGoals(float DeltaTime);
void HandleCarHitBall(class ACar_TA* InCar, class ABall_TA* InBall);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleDemolish(class AGameEvent_TA* InGameEvent, struct FDemolishData Data);
float GetScoreSeparation();
void UpdateScoreState();
void UpdateStates();
void TickActive(float DeltaTime);
void AfterGoalScored();
void GoalScored();
void HandleGameStateChanged(class AGameEvent_TA* G);
void ClearPendingCrowdFlags();
void InitModifiers();
void InitGameObserver();
void eventDetached();
void eventTickInit();
};

// Class TAGame.CrowdSoundSettings_TA
// 0x0160 (0x0100 - 0x0260)
class UCrowdSoundSettings_TA : public UCrowdSoundSettingsBase_TA
{
public:
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionScorability; // 0x0100
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionOpenGoal; // 0x0108
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionSpeedTowardsGoal; // 0x0110
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionCloseToGoal; // 0x0118
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionDistanceFromGoal; // 0x0120
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionAerial; // 0x0128
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERDURATION_X* TensionFastBreak; // 0x0130
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionTimeSinceLastGoal; // 0x0138
    (0x0008) [0x0001000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink |
    CPF_EditInline)
    class UDYNAMICVALUEMODIFIERCURVE_X* TensionScoreSeparation; // 0x0140

```

(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   TensionTimeNearGoal;                           // 0x0148  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   TensionLeadChanges;                           // 0x0150  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               TensionGoalScored;                           // 0x0158  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               ExcitementKickoffTouch;                   //  
0x0160 (0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   ExcitementScorability;                   // 0x0168  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               ExcitementFastBreak;                   // 0x0170  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               ExcitementGoal;                           // 0x0178  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               ExcitementAerial;                           // 0x0180  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               ExcitementTenseClear;                   // 0x0188  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   ExcitementLeadChanges;                   //  
0x0190 (0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   ExcitementScoreSeparation;                   //  
0x0198 (0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   ExcitementTimeSinceLastGoal;           //  
0x01A0 (0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               UpsetAnyDemo;                           // 0x01A8  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               UpsetRelevantDemo;                   // 0x01B0  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               UpsetGoal;                                   // 0x01B8  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   AnticipationScorability;                   // 0x01C0  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierDuration\_X\*               AnticipationFastBreak;                   // 0x01C8  
(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
class UDynamicValueModifierCurve\_X\*                   AnticipationOpenGoal;                   // 0x01D0

(0x0008) [0x0001000004400009] (CPF\_Edit | CPF\_ExportObject | CPF\_NeedCtorLink | CPF\_EditInline)  
float OpenGoalMinDuration; // 0x01D8 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float OpenGoalMaxDistance; // 0x01DC (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float OpenGoalMaxDefense; // 0x01E0 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float LongTimeNearGoalDuration; // 0x01E4 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float LongTimeNearGoalMaxDistance; // 0x01E8 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float ShotMaxDistanceToGoal; // 0x01EC (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float ShotScorability; // 0x01F0 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float ShotSpeedTowardsGoal; // 0x01F4 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float SaveScorability; // 0x01F8 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float SaveDistanceToGoal; // 0x01FC (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float SaveHitMaxTime; // 0x0200 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float SaveShotMaxTime; // 0x0204 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float FastBreakMinDistanceToGoal; // 0x0208 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float FastBreakMaxDistanceToGoal; // 0x020C (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float FastBreakSpeedTowardsGoal; // 0x0210 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float FastBreakDuration; // 0x0214 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float MissedShotMinScorability; // 0x0218 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float MissedShotMinTension; // 0x021C (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float MissedShotMinLastHitTime; // 0x0220 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float RelevantDemoMaxDistanceToBall; // 0x0224 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float RelevantDemoMaxBallDistanceToGoal; // 0x0228 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float AerialMinValue; // 0x022C (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float AerialMinDuration; // 0x0230 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float AerialMinCarInAirDuration; // 0x0234 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
float AerialCooldown; // 0x0238 (0x0004)  
[0x0001000000000001] (CPF\_Edit)  
int32\_t KickoffScoreSeparation; // 0x023C (0x0004)  
[0x0001000000000001] (CPF\_Edit)

```

float TenseNearGoalMinTension; // 0x0240 (0x0004)
[0x0001000000000001] (CPF_Edit)
float TenseNearGoalDuration; // 0x0244 (0x0004)
[0x0001000000000001] (CPF_Edit)
float TenseNearGoalDistance; // 0x0248 (0x0004)
[0x0001000000000001] (CPF_Edit)
float TenseClearDistance; // 0x024C (0x0004)
[0x0001000000000001] (CPF_Edit)
TArray<struct FCrowdScoreState> ScoreStates; // 0x0250 (0x0010)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrowdSoundSettings_TA");
}

return uClassPointer;
};

};

// Class TAGame.CrowdSounds_TA
// 0x01E4 (0x00FC - 0x02E0)
class UCrowdSounds_TA : public UCrowdSoundsBase_TA
{
public:
struct FCrowdExcitementSounds GoalScored; // 0x0100 (0x0028)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds ShotAttempt; // 0x0128
(0x0028) [0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds OpenGoal; // 0x0150 (0x0028)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds Save; // 0x0178 (0x0028)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds Aerial; // 0x01A0 (0x0028)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds LongTimeNearGoal; // 0x01C8
(0x0028) [0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds FastBreak; // 0x01F0 (0x0028)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds Hit0Second; // 0x0218 (0x0028)
[0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds AnyDemolition; // 0x0240
(0x0028) [0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds RelevantDemolition; // 0x0268
(0x0028) [0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds KickoffTouch; // 0x0290
(0x0028) [0x000100000400001] (CPF>Edit | CPF_NeedCtorLink)
struct FCrowdExcitementSounds TenseClear; // 0x02B8 (0x0028)

```

[0x0001000000400001] (CPF\_Edit | CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.CrowdSounds_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.CrumbTrailMetrics_TA  
// 0x0000 (0x0080 - 0x0080)  
class UCrumbTrailMetrics_TA : public UMetricsGroup_X  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.CrumbTrailMetrics_TA");  
}  
  
return uClassPointer;  
};  
  
void Completed(struct FName CompletedTrail, unsigned long bWasActive, unsigned long  
bRepeatable);  
void Activated(struct FName ActivatedTrail);  
void InitFromRedefinition(TArray<struct FName> ActiveTrails);  
void InitFromLoadedSave(TArray<struct FName> ActiveTrails);  
};  
  
// Class TAGame.CrumbTrailRedefinition_TA  
// 0x0010 (0x0060 - 0x0070)  
class UCrumbTrailRedefinition_TA : public UObject  
{  
public:  
TArray<struct FCrumbRedefinedInfo> CrumbRedefInfo; // 0x0060  
(0x0010) [0x0001000000400001] (CPF_Edit | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrumbTrailRedefinition_TA");
}

return uClassPointer;
};

void DebugPrint();
void RedefineCrumbTrails(TArray<struct FCrumbTrailData>& CrumbTrails);
};

// Class TAGame.CrumbTrailSave_TA
// 0x0020 (0x00C8 - 0x00E8)
class UCrumbTrailSave_TA : public USaveObject_TA
{
public:
TArray<struct FName> CompletedTrails; // 0x00C8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FName> ActiveTrails; // 0x00D8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.CrumbTrailSave_TA");
}

return uClassPointer;
};

void DebugPrint();
void SetTrailActive(struct FName InPathName, unsigned long bRepeatable);
bool IsTrailActive(struct FName InPathName);
bool IsTrailCompleted(struct FName InPathName);
void CompleteTrail(struct FName InPathName);
bool ShouldForceSave();
};

// Class TAGame.StatusTrigger_Named_TA
// 0x0008 (0x0080 - 0x0088)
class UStatusTrigger_Named_TA : public UStatusTrigger_X
{
public:
struct FName ConditionalValue; // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:

```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_Named_TA");
}

return uClassPointer;
};

void DebugPrint();
void HandleChange(struct FName InValue);
};

// Class TAGame.StatusTrigger_MenuButton_TA
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_MenuButton_TA : public UStatusTrigger_Named_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_MenuButton_TA");
}

return uClassPointer;
};

};

// Class TAGame.MenuTreeNodeCrumbTrail_TA
// 0x0020 (0x0060 - 0x0080)
class UMenuTreeNodeCrumbTrail_TA : public UObject
{
public:
unsigned long          bTrailEnabled : 1;           // 0x0060 (0x0004)
[0x0001004000002000] [0x00000001] (CPF_Transient)
struct FScriptDelegate      __EventTrailUpdated__Delegate;    // 0x0068
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNodeCrumbTrail_TA");
}

return uClassPointer;
};

bool __MenuTreeNodeCrumbTrail_TA_SetTrailEnabled_0x1(class UMenuTreeNode_TA* N);
void SetTrailEnabled(unsigned long bNewEnabled);
void CompleteTrail();
void ActivateTrail();
bool IsTrailActive();
void EventTrailUpdated();
};

// Class TAGame.CustomMatchSettingsSave_TA
// 0x0098 (0x00C8 - 0x0160)
class UCustomMatchSettingsSave_TA : public USaveObject_TA
{
public:
    struct FCustomMatchSettings           Settings;           // 0x00C8 (0x0090)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class UGameTags_TA*                  GameTags;          // 0x0158 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.CustomMatchSettingsSave_TA");
        }
    }

    return uClassPointer;
};

struct FCustomMatchSettings GetSettings();
void OnLoad();
};

// Class TAGame.DebugHelpMenu_TA
// 0x0038 (0x00D0 - 0x0108)
class UDebugHelpMenu_TA : public UIInteraction
{
public:
    float             CategoryTextScale;      // 0x00D0 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float             ItemTextScale;          // 0x00D4 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float             KeyTextScale;          // 0x00D8 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    TArray<struct FDebugCategory>   HelpCategories;      // 0x00E0
    (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

```

```

int32_t SelectedCategoryIdx; // 0x00F0 (0x0004)
[0x0000000000002000] (CPF_Transient)
int32_t SelectedItemIdx; // 0x00F4 (0x0004)
[0x0000000000002000] (CPF_Transient)
float ItemsWidth; // 0x00F8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float ItemsHeight; // 0x00FC (0x0004)
[0x0000000000002000] (CPF_Transient)
float Fade; // 0x0100 (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long bClosing : 1; // 0x0104 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DebugHelpMenu_TA");
}

return uClassPointer;
};

void Close();
void ExecuteItem(int32_t CatIdx, int32_t ItemIdx);
bool HandleInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float AmountDepressed, unsigned long bGamepad);
void DrawItem(class UCanvas* C, int32_t CategoryIdx, int32_t ItemIdx);
void DrawCategory(class UCanvas* C, int32_t CategoryIdx);
void eventPostRender(class UCanvas* C);
void BuildHelpItems();
int32_t GetCategoryIndex(class FString CategoryName);
bool GetHelpValues(class FString Cmd, class FString& Command, class FString& Category, class FString& Description);
};

// Class TAGame.DebugInput_TA
// 0x0000 (0x02F8 - 0x02F8)
class UDebugInput_TA : public UPlayerInput
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DebugInput_TA");
}
}

```

```
return uClassPointer;
};

};

// Class TAGame.DecalBlockingActor_TA
// 0x0000 (0x0288 - 0x0288)
class ADecalBlockingActor_TA : public AStaticMeshActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DecalBlockingActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.DisplayNameResponse
// 0x0010 (0x0060 - 0x0070)
class UDisplayNameResponse : public UObject
{
public:
class FString                               DisplayName;           // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DisplayNameResponse");
}

return uClassPointer;
};

};

// Class TAGame.DynamicLogosConfig_TA
// 0x0088 (0x0078 - 0x0100)
class UDynamicLogosConfig_TA : public UOnlineConfig_X
{
```

```

public:
unsigned long          bUseDynamicLogos : 1;           // 0x0078 (0x0004)
[0x0001000000000000] [0x00000001]
class FString          LogoURL;                  // 0x0080 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
class FString          FallbackLogoAssetPath;    // 0x0090 (0x0010)
[0x0001084000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
class FString          FallbackLogoAssetPathJPN; // 0x00A0 (0x0010)
[0x0001084000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)
class UTexture*        LogolImage;               // 0x00B0 (0x0008)
[0x0001004000000000]
class FString          LogoLanguage;            // 0x00B8 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
float                 MaxWaitLogoLoadTime;      // 0x00C8 (0x0004)
[0x0001000000000000]
struct FScriptDelegate _EventLogolImageChanged__Delegate; // 0x00D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventLoadFallbackLogo__Delegate; // 0x00E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DynamicLogosConfig_TA");
}

return uClassPointer;
};

bool ShouldSyncLogoFromOnline();
void SetLogoLanguage(class FString InLanguage);
class FString GetLogoLanguage();
class UTexture* GetFallbackLogo();
class UTexture* GetLogolImage();
bool IsDynamicLogosEnabled();
bool IsDynamicLogoLoaded();
void SetLogolImage(class UTexture* InLogolImage);
void HandleLogoSynced(class UTexture* InLogolImage);
void SyncLogoFromOnline();
void Apply();
void EventLoadFallbackLogo();
void EventLogolImageChanged(class UTexture* InLogolImage);
};

// Class TAGame.DynamicMapEventsConfig_TA
// 0x0018 (0x0078 - 0x0090)
class UDynamicMapEventsConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bEnabled : 1;           // 0x0078 (0x0004)

```

```
[0x0001000000000001] [0x00000001] (CPF_Edit)
TArray<class FString> ActiveDynamicMapEvents; // 0x0080
(0x0010) [0x000100000402001] (CPF_Edit | CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DynamicMapEventsConfig_TA");
}

return uClassPointer;
};

void ValidateDynamicMapEvents();
void Apply();
};

// Class TAGame.SeqEvent_DynamicMapEvents_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_DynamicMapEvents_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_DynamicMapEvents_TA");
}

return uClassPointer;
};

bool DynamicMapEventsUpdated();
};

// Class TAGame.DynamicMeshActor_TA
// 0x0008 (0x0268 - 0x0270)
class ADynamicMeshActor_TA : public AActor
{
public:
class UStaticMeshComponent* StaticMeshComponent; // 0x0268
(0x0008) [0x0000000040A000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_EditConst | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.DynamicMeshActor_TA");  
}  
  
return uClassPointer;  
};  
  
void ToggleBallPosition(unsigned long bAdd);  
void eventDestroyForMapUnload();  
void eventDestroyed();  
void eventPostBeginPlay();  
};  
  
// Class TAGame.PlayerStart_TA  
// 0x0000 (0x0398 - 0x0398)  
class APlayerStart_TA : public APlayerStart  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.PlayerStart_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.DynamicPlayerStart_TA  
// 0x0000 (0x0398 - 0x0398)  
class ADynamicPlayerStart_TA : public APlayerStart_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.DynamicPlayerStart_TA");  
}
```

```

return uClassPointer;
};

};

// Class TAGame.DynamicSpawnPoint_TA
// 0x0000 (0x0268 - 0x0268)
class ADynamicSpawnPoint_TA : public AActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DynamicSpawnPoint_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameEditor_Actor_TA
// 0x0048 (0x0268 - 0x02B0)
class AGameEditor_Actor_TA : public AActor
{
public:
class UPrimitiveComponent* Mesh; // 0x0268 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FVector RotateCameraOffset; // 0x0270 (0x000C)
[0x0000000000000000]
float MaxRotatePitch; // 0x027C (0x0004)
[0x0000000000000000]
float MaxInputRotatePitch; // 0x0280 (0x0004)
[0x0000000000000000]
unsigned long bModifyPitchOnEditorSetRotation : 1; // 0x0284
(0x0004) [0x0000000000000000] [0x00000001]
class AFXActor_X* FXActorArchetype; // 0x0288 (0x0008)
[0x0000000000000001] (CPF_Editable)
class AFXActor_X* FXActor; // 0x0290 (0x0008)
[0x0000008000002000] (CPF_Transient)
struct FScriptDelegate __EventSelectionChange__Delegate; // 0x0298
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GameEditor_Actor_TA");
}

return uClassPointer;
};

void EditingEnd();
void EditingBegin();
void NotifyOnSelectionChange(struct FScriptDelegate Callback);
struct FVector EditorGetLocation();
struct FRotator EditorGetRotation();
bool CanEdit();
bool SerializedDataIsEqual(class FString S0, class FString S1);
void Deserialize(class UJsonObject* Data);
class FString Serialize();
struct FVector GetCameraPositionOffset();
struct FVector CollisionExtent();
void ModifyBlendedRotation(struct FRotator ControllerRot, struct FRotator DeltaRot, struct
FRotator& out_Rotation);
struct FVector GetGrabRotationOffset(struct FRotator CurrentRotation);
struct FRotator GetDesiredRotation();
void ReInitPhysics();
void ReInitPhysicsDelayed();
bool EditorSetLocation(struct FVector NewLocation);
bool EditorMoveToLocation(struct FVector NewLocation);
void AddToRotation(struct FRotator Rot, float MaxDegrees);
void EditorSetRotationFromInput(float Forward, float Strafe);
void EditorSetRotation(struct FRotator NewRotation);
void InitFX();
void eventPostBeginPlay();
void EventSelectionChange(class UInterface_GameEditor_TA* Object, unsigned long bSelected);
};

// Class TAGame.DynamicSpawnPointMesh_TA
// 0x0000 (0x02B0 - 0x02B0)
class ADynamicSpawnPointMesh_TA : public AGameEditor_Actor_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.DynamicSpawnPointMesh_TA");
}

return uClassPointer;
};

};

```

```

// Class TAGame.EndRoundComponent_TA
// 0x000C (0x00A4 - 0x00B0)
class UEndRoundComponent_TA : public UActorComponent_X
{
public:
class AGameEvent_Soccar_TA*           SoccarGame;          // 0x00A8
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EndRoundComponent_TA");
}

return uClassPointer;
};

void EndRound();
void HandleBallHitGroundTimeout();
void HandleBallHitGround(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNorm);
void eventDetached();
void eventAttached();
};

// Class TAGame.OvertimeComponent_TA
// 0x0018 (0x00A4 - 0x00BC)
class UOvertimeComponent_TA : public UActorComponent_X
{
public:
uint8_t                  TieBreaker;          // 0x00A8 (0x0001)
[0x0000000000000001] (CPF_Edit)

int32_t                  OvertimeGameTime;    // 0x00AC (0x0004)
[0x0000000000000001] (CPF_Edit)

float                   BallHitGroundTimeoutSeconds; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)

int32_t                  FirstScoreTeamIndex; // 0x00B4 (0x0004)
[0x0000000000000000]

int32_t                  RandomWinningTeamIndex; // 0x00B8 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OvertimeComponent_TA");
}
}

```

```

return uClassPointer;
};

class ATeam_TA* PickRandomTeam();
class ATeam_TA* PickFirstScore();
class ATeam_TA* PickWinner();
void HandleGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class
UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
void eventDetached();
void eventAttached();
};

// Class TAGame.EndRoundComponent_HauntedBall_TA
// 0x0000 (0x00B0 - 0x00B0)
class UEndRoundComponent_HauntedBall_TA : public UEndRoundComponent_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EndRoundComponent_HauntedBall_TA");
}

return uClassPointer;
};

void HandleBallHitGround(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNorm);

// Class TAGame.EndRoundComponent_KnockOut_TA
// 0x0000 (0x00B0 - 0x00B0)
class UEndRoundComponent_KnockOut_TA : public UEndRoundComponent_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EndRoundComponent_KnockOut_TA");
}

return uClassPointer;
};

```

```
void eventAttached();
};

// Class TAGame.EndRoundComponent_Rugby_TA
// 0x0000 (0x00B0 - 0x00B0)
class UEndRoundComponent_Rugby_TA : public UEndRoundComponent_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EndRoundComponent_Rugby_TA");
}

return uClassPointer;
};

void HandleBallHitGround(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNorm);
};

// Class TAGame.EndRoundComponent_TrainingEditor_TA
// 0x0000 (0x00B0 - 0x00B0)
class UEndRoundComponent_TrainingEditor_TA : public UEndRoundComponent_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EndRoundComponent_TrainingEditor_TA");
}

return uClassPointer;
};

void HandleBallHitGroundTimeout();
void HandleBallHitGround(class ABall_TA* Ball, struct FVector HitLoc, struct FVector HitNorm);
};

// Class TAGame.EndRoundComponent_Tutorial_TA
// 0x0000 (0x00B0 - 0x00B0)
class UEndRoundComponent_Tutorial_TA : public UEndRoundComponent_TA
{
public:
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EndRoundComponent_Tutorial_TA");
}

return uClassPointer;
};

void EndRound();
};

// Class TAGame.EngagementEventNotification_TA
// 0x0010 (0x0170 - 0x0180)
class UEngagementEventNotification_TA : public UNotification_TA
{
public:
class FString           UIEngagementEventTypeAnim;          // 0x0170 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EngagementEventNotification_TA");
}

return uClassPointer;
};

class UEngagementEventNotification_TA* SetAnim(class FString InAnim);
};

// Class TAGame.GFxData_EngagementEventNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_EngagementEventNotification_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
```

```
TAGame.GFxData_EngagementEventNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.SystemSettingsManager_TA
// 0x0004 (0x0060 - 0x0064)
class USystemSettingsManager_TA : public UObject
{
public:
int32_t NumPlayers; // 0x0060 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SystemSettingsManager_TA");
}

return uClassPointer;
};

void ApplySettings(class UGameViewportClient_TA* GVC);
void UpdateNumPlayers(class UGameViewportClient_TA* GVC);
void HandleSplitScreenTypeChanged(class UGameViewportClient_TA* GVC);
void HandleWorldFullScreenChanged(class UGameViewportClient_TA* GVC);
void Init();
};

// Class TAGame.EnteredMainMenu_TA
// 0x0000 (0x0060 - 0x0060)
class UEnteredMainMenu_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EnteredMainMenu_TA");
}

return uClassPointer;
};
```

```

};

// Class TAGame.EnvironmentZone_TA
// 0x0000 (0x0060 - 0x0060)
class UEnvironmentZone_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EnvironmentZone_TA");
}

return uClassPointer;
};

int32_t GetEnvironmentZone();
};

// Class TAGame.EOSGameClipsConfig_TA
// 0x0038 (0x0078 - 0x00B0)
class UEOSGameClipsConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bFeatureEnabled : 1;           // 0x0078 (0x0004)
[0x0009000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bCabinModeClippingEnabled : 1;    // 0x0078
(0x0004) [0x00100000000000] [0x00000002]
unsigned long          bGameplayClippingEnabled : 1;     // 0x0078 (0x0004)
[0x00100000000000] [0x00000004]
unsigned long          bGoalReplayClippingEnabled : 1;   // 0x0078 (0x0004)
[0x00100000000000] [0x00000008]
unsigned long          bLinkAccountPopupEnabled : 1;     // 0x0078 (0x0004)
[0x00100000000000] [0x00000010]
unsigned long          bSettingsMenuButtonEnabled : 1;   // 0x0078 (0x0004)
[0x00100000000000] [0x00000020]
unsigned long          bStaticDataDebugEnabled : 1;     // 0x0078 (0x0004)
[0x00100004000000] [0x00000040] (CPF_EditInlineNotify)
uint64_t               ClipDurationSeconds;           // 0x0080 (0x0008)
[0x00100000000000]
float                 MinClipLengthSeconds;           // 0x0088 (0x0004)
[0x00100000000000]
float                 MinClipCooldownSeconds;          // 0x008C (0x0004)
[0x00100000000000]
float                 MinClipCooldownAfterRecordingStartedSeconds; // 0x0090
(0x0004) [0x00100000000000]
int32_t               MaxClipUploadsPerMinute;        // 0x0094 (0x0004)
[0x00100000000000]

```

```

struct FScriptDelegate           __bFeatureEnabled__ChangeNotify;      // 0x0098
(0x0018) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSGameClipsConfig_TA");
}

return uClassPointer;
};

void __bFeatureEnabled__ChangeNotifyFunc();
};

// Class TAGame.EOSGameClipsController_TA
// 0x0088 (0x0060 - 0x00E8)
class UEOSGameClipsController_TA : public UObject
{
public:
class UOnlineGameClipsInterface*          GameClipsInterface_Object;    // 0x0060
(0x0008) [0x0001004000000000]
class UOnlineGameClipsInterface*          GameClipsInterface_Interface; // 0x0068
(0x0008) [0x0001004000000000]
class UEOSGameClipsManager_TA*            GameClipsManager;           // 0x0070
(0x0008) [0x0001800000000000]
class UEOSGameClipsConfig_TA*            GameClipsConfig;            // 0x0078
(0x0008) [0x0001800000000000]
class UAkSoundCue*                      ClipCaptured_SFX;          // 0x0080 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UAkSoundCue*                      ClipFailed_SFX;            // 0x0088 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UAkSoundCue*                      ClipSucceeded_SFX;         // 0x0090 (0x0008)
[0x0001000000000001] (CPF_Edit)
unsigned long                           bGoalReplayClipInitiated : 1; // 0x0098 (0x0004)
[0x0001000000000000] [0x00000001]
float                                  LastClipTimeSeconds;        // 0x009C (0x0004)
[0x0001000000000000]
struct FScriptDelegate           __EventClippedDuringReplay__Delegate; // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventClipStatusChanged__Delegate;  // 0x00B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate           __EventErrorOccurred__Delegate;   // 0x00D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSGameClipsController_TA");
}

return uClassPointer;
};

void __EOSGameClipsController_TA__HandleGameClipsManagerChanged_0x1();
void __EOSGameClipsController_TA__SetGameClipsInterface_0x2(class FString InEpicAccountId,
int32_t InClipId, class UErrorType* InErrorType);
void __EOSGameClipsController_TA__SetGameClipsInterface_0x1(class UErrorType*
InErrorType);
void PlaySoundEffect(class UAkSoundCue* SoundEffect);
class FString GetEpicAccountId();
bool IsAccountLinked();
void CreateClip(class FString InGameClipInputType);
float GetCooldownTime();
float GetTimeUntilUnthrottled();
void CaptureClipPressed(class FString InGameClipInputType);
void HandleErrorOccurred(class FString InEpicAccountId, class UErrorType* InErrorType);
void HandleClipStatusChanged(class FString InEpicAccountId, int32_t InClipId, uint8_t
InNewClipStatus);
void HandleRecordingChanged(uint8_t InNewRecording);
void SetGameClipsInterface(class UOnlineGameClipsInterface* InGameClipsInterface);
void HandleGameClipsManagerChanged();
void ShutDown();
void eventConstruct();
void EventErrorOccurred(class UErrorType* InErrorType);
void EventClipStatusChanged(int32_t InClipId, uint8_t InNewClipStatus);
void EventClippedDuringReplay();
};

// Class TAGame.EOSGameClipsManager_TA
// 0x0100 (0x0060 - 0x0160)
class UEOSGameClipsManager_TA : public UObject
{
public:
class UOnlineGameClipsInterface* GameClipsInterface_Object; // 0x0060
(0x0008) [0x0001004000000000]
class UOnlineGameClipsInterface* GameClipsInterface_Interface; // 0x0068
(0x0008) [0x0001004000000000]
class UEOSGameClipsConfig_TA* GameClipsConfig; // 0x0070
(0x0008) [0x0001800000000000]
class AGameEvent_TA* CurrentGameEvent; // 0x0078 (0x0008)
[0x0001800000000000]
class FString ClipType_Gameplay; // 0x0080 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
class FString ClipType_GoalReplay; // 0x0090 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
class FString ClipType_None; // 0x00A0 (0x0010)
[0x0001004000400002] (CPF_Const | CPF_NeedCtorLink)
class FString CurrentClipType; // 0x00B0 (0x0010)
[0x0009004000400000] (CPF_NeedCtorLink)

```

```

class FString PreviousClipType; // 0x00C0 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
unsigned long bClippingAvailable : 1; // 0x00D0 (0x0004)
[0x0009004000000000] [0x00000001]
unsigned long bEnableDebug : 1; // 0x00D0 (0x0004)
[0x0010000000000000] [0x00000002]
TArray<class FString> LinkedAccounts; // 0x00D8 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
TArray<class FString> InactiveAccounts; // 0x00E8 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
float MostRecentRecordingStartTime; // 0x00F8 (0x0004)
[0x0001000000000000]
struct FScriptDelegate __EventCreateGameClipInitiated__Delegate; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventMaskStatusChanged__Delegate; // 0x0118
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __CurrentClipType__ChangeNotify; // 0x0130
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __bClippingAvailable__ChangeNotify; // 0x0148
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSGameClipsManager_TA");
}

return uClassPointer;
};

void __EOSGameClipsManager_TA__HandleGameEventChanged_0x1(class AGameEvent_TA* InGameEvent);
void __EOSGameClipsManager_TA__UpdateRecording_0x1();
void __bClippingAvailable__ChangeNotifyFunc();
void __CurrentClipType__ChangeNotifyFunc();
static bool IsSimulating();
bool IsCabinedClippingDisabled();
void UpdateRecording();
void HideMaskArea(uint64_t InMaskId);
uint64_t ShowMaskArea(struct FGameClipsMaskArea InMaskArea);
bool IsClippingInGoalReplay(class FString InGameClipInputType);
bool IsClippingInGameplay(class FString InGameClipInputType);
bool IsClippingInCorrectState(class FString InGameClipInputType);
bool IsAnyAccountActive();
bool IsAnyAccountLinked();
bool IsAccountLinked(class FString InAccountId);
void RemoveInactiveAccount(class FString InAccountId);
void AddInactiveAccount(class FString InAccountId);
void HandleGameStateChanged(struct FName InStateName);
void HandleGameEventChanged();

```

```

void HandleGameClipsConfigChanged();
void HandleMaskStatusChanged(uint64_t InMaskAreaHandle, struct FGameClipsMaskArea
InMaskArea, uint8_t InNewMaskStatus);
void HandleConnectionStatusChanged(class FString InEpicAccountId, uint8_t InConnection,
uint8_t InNewConnectionStatus);
void HandleRecordingChanged(uint8_t NewRecording);
void HandleAvailabilityChanged(uint8_t NewAvailability);
void SetGameClipsInterface(class UOnlineGameClipsInterface* InGameClipsInterface);
void eventConstruct();
void EventMaskStatusChanged(uint64_t InMaskAreaHandle, struct FGameClipsMaskArea
InMaskArea, uint8_t InNewMaskStatus);
void EventCreateGameClipInitiated(class FString InEpicAccountId, int32_t InClipId, class FString
InClipType);
};

// Class TAGame.EOSGameClipsSettingsSave_TA
// 0x0004 (0x00C8 - 0x00CC)
class UEOSGameClipsSettingsSave_TA : public USaveObject_TA
{
public:
unsigned long           bLinkAccountPopupSeen : 1;           // 0x00C8 (0x0004)
[0x0001000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSGameClipsSettingsSave_TA");
}

return uClassPointer;
};

};

// Class TAGame.EOSShopPurchaseEvent_X
// 0x0028 (0x0068 - 0x0090)
class UEOSShopPurchaseEvent_X : public UEOSMetricEvent_X
{
public:
class FString           CurrencyType;           // 0x0068 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t                 CurrencyAmount;         // 0x0078 (0x0004)
[0x0001000000000000]

TArray<struct FEOSShopPurchaseEventProduct>   Products;           // 0x0080
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSShopPurchaseEvent_X");
}

return uClassPointer;
};

};

// Class TAGame.EOSMtxPurchaseEvent_X
// 0x0000 (0x0090 - 0x0090)
class UEOSMtxPurchaseEvent_X : public UEOSShopPurchaseEvent_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSMtxPurchaseEvent_X");
}

return uClassPointer;
};

};

// Class TAGame.EOSOnlineStorageSync_TA
// 0x0028 (0x0060 - 0x0088)
class UEOSOnlineStorageSync_TA : public UObject
{
public:
class UAsyncTask*           SyncTask;           // 0x0060 (0x0008)
[0x0000000000000000]
int32_t                     EOS_FILE_CORRUPTED; // 0x0068 (0x0004)
[0x0000000000000002] (CPF_Const)
struct FScriptDelegate       __EventSyncObjectSuccess__Delegate; // 0x0070
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSOnlineStorageSync_TA");
}

```

```

return uClassPointer;
};

void HandleFileReadComplete(unsigned long bWasSuccessful, class FString UserId, class
FString Filename);
void HandleEnumerateUserFilesComplete(unsigned long bWasSuccessful, class FString UserId);
class UAsyncTask* SyncCloudSave();
void Unload();
void EventSyncObjectSuccess(class UObject* RemoteObject);
};

// Class TAGame.OnlineStorageConfig_TA
// 0x002C (0x0078 - 0x00A4)
class UOnlineStorageConfig_TA : public UOnlineConfig_X
{
public:
TArray<class UClass*> DisabledClasses; // 0x0078 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString OnlineStorageFileName; // 0x0088 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
uint64_t MinSecondsBetweenCloudUploads; // 0x0098 (0x0008)
[0x0000000000000000]
float UploadRetryTime; // 0x00A0 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineStorageConfig_TA");
}

return uClassPointer;
};

class FString GetOnlineStorageFileName();
};

// Class TAGame.EOSOnlineStorageUploader_TA
// 0x0030 (0x0060 - 0x0090)
class UEOSOnlineStorageUploader_TA : public UObject
{
public:
class UOnlineStorageConfig_TA* Config; // 0x0060 (0x0008)
[0x0000800000000001] (CPF_Edit)
unsigned long bSaveDataDirty : 1; // 0x0068 (0x0004)
[0x0000080000000000] [0x00000001]
unsigned long bUploadInFlight : 1; // 0x0068 (0x0004)
[0x0000080000000000] [0x00000002]
uint64_t LastSuccessfulUploadTime; // 0x0070 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```

struct FScriptDelegate           __EventSaveUploadComplete__Delegate;      // 0x0078
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSOnlineStorageUploader_TA");
}

return uClassPointer;
};

struct FOnlineSaveObject __EOSOnlineStorageUploader_TA__UploadSaveData_0x2(class
USaveObject_TA* S);
bool __EOSOnlineStorageUploader_TA__UploadSaveData_0x1(class USaveObject_TA* S);
void HandleFileWriteComplete(unsigned long bWasSuccessful, class FString UserId, class
FString Filename);
bool UploadSaveData(unsigned long bForce);
void UploadSaveDataTimer();
void SetSaveDataDirty();
void Unload();
void HandleEOSInitialized();
void eventConstruct();
void EventSaveUploadComplete(unsigned long bWasSuccessful, class FString UserId, class
FString Filename);
};

// Class TAGame.EOSPermissions_TA
// 0x00A8 (0x0060 - 0x0108)
class UEOSPermissions_TA : public UObject
{
public:
class UEpicConfig_X*          EpicConfig;                      // 0x0060 (0x0008)
[0x0000800000000000]
class UPsyNetRetryConfig_X*    RetryConfig;                    // 0x0068 (0x0008)
[0x0000800000000000]
struct FPlayerPermissions       Permissions;                  // 0x0070 (0x0008)
[0x0000004000000000]
class UGameplaySettingsSave_TA* GameplaySettingsSave;        // 0x0078
(0x0008) [0x0000004000000000]
class UNotificationSave_TA*    NotificationSave;            // 0x0080 (0x0008)
[0x0000004000000000]
class UK3SUserPermissions_TA*   DownloadedPermissions;        // 0x0088
(0x0008) [0x0000000000000000]
class FString                  ServiceName;                // 0x0090 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
TArray<float>                 RetryDelays;                // 0x00A0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t                        PermissionRetryAttempts;     // 0x00B0 (0x0004)
[0x0000000000000000]

```

```
class UWebRequest_X*           ActivePermissionsRequest;          // 0x00B8
(0x0008) [0x0000000000000000] (CPF_Transient)
struct FScriptDelegate         __EventPlayerPermissionsSet__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __EventSettingsAutoUpdated__Delegate; // 0x00D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate         __ConvertStringDelegate__Delegate;   // 0x00F0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSPermissions_TA");
}

return uClassPointer;
};

void __EOSPermissions_TA__Construct_0x1(class USaveGameManager_TA* M, class
USaveData_TA* S, class UError* E);
void __EOSPermissions_TA__UpdateSaveObjectReferences_0x3(class UObject* SaveObject);
void __EOSPermissions_TA__UpdateSaveObjectReferences_0x2(class UObject* SaveObject);
void __EOSPermissions_TA__UpdateSaveObjectReferences_0x1(class UObject* SaveObject);
void __EOSPermissions_TA__RequestEOSPermissions_0x1(unsigned long bSuccess, class
FString EpicAuthTicket);
void __EOSPermissions_TA__RequestEOSPermissions_0x2(class UWebRequest_X* Response);
class FString __EOSPermissions_TA__UpdatePlayerPermissions_0x4(class FString S);
class FString __EOSPermissions_TA__UpdatePlayerPermissions_0x3(class FString S);
class FString __EOSPermissions_TA__UpdatePlayerPermissions_0x2(class FString S);
class FString __EOSPermissions_TA__UpdatePlayerPermissions_0x1(class FString S);
static uint8_t GetChatFilterForPermLevel(uint8_t PermissionLevel, uint8_t DefaultValue);
void NotifyOnDownloadedPermissions(struct FScriptDelegate Callback);
bool GetAreShopNotificationsEnabledDefault();
bool AreShopNotificationsEnabled();
void SetEnforceCabinedMode();
bool EnforceCabinedMode();
bool PromptForPin();
void SetRequirePinForFriends(unsigned long bVal);
bool IsPinRequiredForFriends();
uint8_t GetVoiceChatPermissionDefaultLevel();
uint8_t GetTextChatPermissionDefaultLevel();
uint8_t GetVoiceChatPermissionLevel();
uint8_t GetTextChatPermissionLevel();
bool GetMostRestrictiveBoolPermission(struct FK3SSetting Setting, unsigned long
bRestrictiveValue);
uint8_t GetMostRestrictiveChatPermission(struct FK3SSetting Setting);
void OverrideSaveDataSettings();
void OnPermissionCheckFailed(class UError* InError);
class FString DoStringConversion(class FString InValue, struct FScriptDelegate
ConversionDelegate);
```

```

void LogPermission(struct FK3SSetting Setting, class FString InterpretedValue, struct
FScriptDelegate ConversionDelegate);
void UpdatePlayerPermissions();
void RequestEOSPermissions();
void HandlePsyNetConnected(class UPsyNetConnection_X* C);
void CopyPrimaryPlayerPermissions();
void UpdateSaveObjectReferences();
void eventConstruct();
class FString ConvertStringDelegate(class FString InValue);
void EventSettingsAutoUpdated(uint8_t Reason);
void EventPlayerPermissionsSet();
};

// Class TAGame.K3SUserPermissions_TA
// 0x0070 (0x0060 - 0x00D0)
class UK3SUserPermissions_TA : public UObject
{
public:
struct FK3SResponse           Response;          // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FK3SMeta               Meta;             // 0x0070 (0x0020)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FK3SFirstTimeDefaults FirstTimeDefaults; // 0x0090 (0x0040)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.K3SUserPermissions_TA");
}

return uClassPointer;
};

};

// Class TAGame.EOSUserPermissionsError_TA
// 0x0020 (0x0060 - 0x0080)
class UEOSUserPermissionsError_TA : public UObject
{
public:
class FString           ErrorCode;        // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString           Message;          // 0x0070 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSUserPermissionsError_TA");
}

return uClassPointer;
};

};

// Class TAGame.EOSUserPermissions_TA
// 0x0138 (0x0060 - 0x0198)
class UEOSUserPermissions_TA : public UObject
{
public:
struct FEOSPermissionsResponse           Data;                      // 0x0060 (0x0138)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSUserPermissions_TA");
}

return uClassPointer;
};

};

// Class TAGame.EOSVoiceConfig_TA
// 0x0028 (0x0078 - 0x00A0)
class UEOSVoiceConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bEnabled : 1;                      // 0x0078 (0x0004)
[0x0009000000000000] [0x00000001]
float                  MatchRequestDelay;                // 0x007C (0x0004)
[0x0001000000000000]
int32_t                TextNotificationCooldown;        // 0x0080 (0x0004)
[0x0001000000000000]
struct FScriptDelegate      _bEnabled__ChangeNotify;    // 0x0088
(0x0018) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.EOSVoiceConfig_TA");
}

return uClassPointer;
};

void __bEnabled__ChangeNotifyFunc();
};

// Class TAGame.EOSVoiceErrors_TA
// 0x00E0 (0x0080 - 0x0160)
class UEOSVoiceErrors_TA : public UErrorList
{
public:
class UErrorType* VoiceError; // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceChatBanned; // 0x0088 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceNotPrimaryPlayer; // 0x0090 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceNoPlatformPermissions; // 0x0098
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* VoiceNoEpicPermissions; // 0x00A0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceNoSpectatorPermissions; // 0x00A8
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* VoiceTokenRequestDenied; // 0x00B0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceDisabled; // 0x00B8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceSettingDisabled; // 0x00C0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceInvalidPartyToken; // 0x00C8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceAlreadyJoiningRoom; // 0x00D0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceAlreadyInRoom; // 0x00D8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceRoomNotValid; // 0x00E0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceLeaveFailed; // 0x00E8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceRemoteUserNotAllowed; // 0x00F0
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* VoiceRemoteUserJoined; // 0x00F8 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceNoConnection; // 0x0100 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* VoiceJoinFailedTooManyParticipants; // 0x0108
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* VoiceJoinFailedInvalidCredentials; // 0x0110
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType* VoiceJoinFailed; // 0x0118 (0x0008)
}

```

```

[0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceUnexpectedError;          // 0x0120 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceAlreadyRequestingToken; // 0x0128
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceDisabledForTeam;        // 0x0130 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceCrossplatformChatNotEnabled; // 0x0138
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           VoicePlayerPlatformMuted;    // 0x0140 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceCredentialsNotValid;    // 0x0148 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceRoomNotFound;          // 0x0150 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           VoiceRemovedDueToSettingChange; // 0x0158
(0x0008) [0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EOSVoiceErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_RequestPartyVoiceToken_TA
// 0x00C8 (0x00E8 - 0x01B0)
class URPC_RequestPartyVoiceToken_TA : public URPC_X
{
public:
struct FUniqueNetId           RequestingPlayer;          // 0x00E8 (0x0048)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FUniqueNetId           PlayerID;                // 0x0130 (0x0048)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                 RoomId;                  // 0x0178 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                 BaseUrl;                 // 0x0188 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                 Token;                   // 0x0198 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UBanMessage_X*          BanMessage;              // 0x01A8 (0x0008)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RequestPartyVoiceToken_TA");
}

return uClassPointer;
};

class URPC_RequestPartyVoiceToken_TA* SetRequestingPlayer(struct FUniqueNetId
InRequestingPlayer);
};

// Class TAGame.PartyMessage_CurrentVoiceRoom_TA
// 0x0058 (0x00A8 - 0x0100)
class UPartyMessage_CurrentVoiceRoom_TA : public UPartyMessage_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00A8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 RoomName;          // 0x00F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_CurrentVoiceRoom_TA");
}

return uClassPointer;
};

class UPartyMessage_CurrentVoiceRoom_TA* SetRoomName(class FString InRoomName);
class UPartyMessage_CurrentVoiceRoom_TA* SetPlayerID(struct FUniqueNetId& InPlayerId);
};

// Class TAGame.EOSVoiceMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UEOSVoiceMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.EOSVoiceMetrics_TA");
}

return uClassPointer;
};

void VoiceRoomActivitySummary(struct FUniqueNetId& PlayerID, struct FVoiceRoomDetails&
RoomDetails, struct FVoiceRoomPlayerActivity& PlayerActivity);
void VoiceSettings(struct FUniqueNetId& PlayerID, struct FUserVoiceSettings& Settings);
void RecordUserVoiceSettings(struct FUniqueNetId PlayerID, class UEOSVoiceSettingsSave_TA*
VoiceSave);
};

// Class TAGame.RPC_RequestMatchVoiceTokens_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_RequestMatchVoiceTokens_TA : public URPC_X
{
public:
TArray<struct FVoiceRoomTokenRequest> RoomTokenRequests; // 0x00E8 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FPlayerVoiceRoomCredentials> PlayerData; // 0x00F8 (0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RequestMatchVoiceTokens_TA");
}

return uClassPointer;
};

class URPC_RequestMatchVoiceTokens_TA* SetRoomTokenRequests(TArray<struct FVoiceRoomTokenRequest>& InRoomTokenRequests);
};

// Class TAGame.EpicFriendsConverterConfig_TA
// 0x0008 (0x0078 - 0x0080)
class UEpicFriendsConverterConfig_TA : public UOnlineConfig_X
{
public:
float SecondsBetweenInvites; // 0x0078 (0x0004) [0x0000000000000001] (CPF_Edit)
int32_t MaxNumberOfAttemptsPerUser; // 0x007C (0x0004) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EpicFriendsConverterConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.UISavedValues_TA
// 0x0010 (0x00C8 - 0x00D8)
class UUISavedValues_TA : public USaveObject_TA
{
public:
TArray<struct FUISavedKeyValue>           Values;           // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UISavedValues_TA");
}

return uClassPointer;
};

void Clear(struct FName Key);
void Set(struct FName Key, class FString Value);
bool Contains(struct FName Key);
class FString Get(struct FName Key);
};

// Class TAGame.EulaConfig_TA
// 0x0028 (0x0078 - 0x00A0)
class UEulaConfig_TA : public UOnlineConfig_X
{
public:
unsigned long           bRequiresAcceptance : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long           bAllowEulaReset : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
TArray<uint8_t>         ExcludedPlatforms;           // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<int32_t>          IgnoredHashIds;           // 0x0090 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EulaConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.EulaSave_TA
// 0x0038 (0x00C8 - 0x0100)
class UEulaSave_TA : public USaveObject_TA
{
public:
TArray<struct FEulaVersion> AcceptedEulas; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t ResetCounter; // 0x00D8 (0x0004)
[0x0000000000000000]
class FString LatestCDNEula; // 0x00E0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString LatestLanguage; // 0x00F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EulaSave_TA");
}

return uClassPointer;
};

void Accepted(int32_t TextHash);
bool RequiresAcceptance(int32_t TextHash);
};

// Class TAGame.EventsPageConfig_TA
// 0x0060 (0x0078 - 0x00D8)
class UEventsPageConfig_TA : public UOnlineConfig_X
{
public:
unsigned long bEnabled : 1; // 0x0078 (0x0004)
[0x0000000000000000] [0x00000001]
class FString URL; // 0x0080 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ButtonTextureURL_Default; // 0x0090 (0x0010)

```

```

[0x0000000000400000] (CPF_NeedCtorLink)
class FString                         ButtonTextureURL_Hover;           // 0x00A0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                         ButtonTextureURL_Click;          // 0x00B0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UTexture*                      ButtonTexture_Default;           // 0x00C0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class UTexture*                      ButtonTexture_Hover;             // 0x00C8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class UTexture*                      ButtonTexture_Click;              // 0x00D0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.EventsPageConfig_TA");
}

return uClassPointer;
};

void __EventsPageConfig_TA__SyncTextures_0x3(class UTexture2DDynamic* Texture);
void __EventsPageConfig_TA__SyncTextures_0x2(class UTexture2DDynamic* Texture);
void __EventsPageConfig_TA__SyncTextures_0x1(class UTexture2DDynamic* Texture);
void SyncTextures();
};

// Class TAGame.ExclusiveInteraction_TA
// 0x0010 (0x00D0 - 0x00E0)
class UExclusiveInteraction_TA : public UInteraction
{
public:
TArray<int32_t>                   ControllersWithExclusiveInputEnabled; // 0x00D0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ExclusiveInteraction_TA");
}

return uClassPointer;
};

bool IsControllerDisabled(int32_t ControllerId);
bool HandleInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float

```

```

AmountDepressed, unsigned long bGamepad);
bool HandleInputChar(int32_t ControllerId, class FString Character);
bool HandleInputAxis(int32_t ControllerId, struct FName Key, float delta, float DeltaTime,
unsigned long bGamepad);
static void EnableExclusiveInput(int32_t ControllerId);
static void DisableExclusiveInput(int32_t ControllerId);
};

// Class TAGame.ExhibitionMatchSettingsSave_TA
// 0x0000 (0x0160 - 0x0160)
class UExhibitionMatchSettingsSave_TA : public UCustomMatchSettingsSave_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ExhibitionMatchSettingsSave_TA");
}

return uClassPointer;
};

};

// Class TAGame.ExplosionPreviewer_TA
// 0x0038 (0x0070 - 0x00A8)
class UExplosionPreviewer_TA : public UComponent
{
public:
class UProductLoader_TA*           ProductLoader;           // 0x0070 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
struct FProductInstanceId          ExplosionId;          // 0x0078 (0x0010)
[0x0000000000002000] (CPF_Transient)
class UProductAsset_GoalExplosion_TA*   ExplosionProduct;    // 0x0088
(0x0008) [0x0000000000002000] (CPF_Transient)
class AFXActor_X*                  ExplosionInstance;      // 0x0090 (0x0008)
[0x0000000000002000] (CPF_Transient)
unsigned long                      bPreviewing : 1;       // 0x0098 (0x0004)
[0x0000000000000000] [0x00000001]
class APlayerController*           PC;                   // 0x00A0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.ExplosionPreviewer_TA");
}

return uClassPointer;
};

void ClearPreview();
void ApplyExplosionParameters(class AFXActor_X* Instance);
void StopExplosion();
void PlayExplosion();
void HandleProductLoaded(class UProductLoader_TA* InLoader);
void SetProduct(int32_t ProductID, struct FProductInstanceId OnlineProductId);
void SetLoadout(struct FLoadoutData& InLoadout);
void UpdateExplosion();
void SetPreviewSlot(class UProductSlot_TA* InSlot);
void SetPreviewSlotIndex(int32_t InSlotIndex);
};

// Class TAGame.SeqEvent_PreviewGoalExplosion_TA
// 0x000C (0x017C - 0x0188)
class USeqEvent_PreviewGoalExplosion_TA : public USequenceEvent
{
public:
    class AFXActor_X*           FXInstance;           // 0x0180 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_PreviewGoalExplosion_TA");
        }
    }

    return uClassPointer;
};

};

// Class TAGame.PaintedFXSelector_TA
// 0x0000 (0x0060 - 0x0060)
class UPaintedFXSelector_TA : public UObject
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {

```

```

uClassPointer = UObject::FindClass("Class TAGame.PaintedFXSelector_TA");
}

return uClassPointer;
};

static bool __PaintedFXSelector_TA__PickFXActor_0x1(class UProductAttribute_TA* X);
static class AFXActor_X* GetFXActorForPRI(class APRI_TA* PRI, int32_t SlotIndex, class
AFXActor_X* PaintedFXActor, class AFXActor_X* FXActor);
static class AFXActor_X* GetFXActorForMesh(class UCarMeshComponentBase_TA* CarMesh,
int32_t SlotIndex, class AFXActor_X* PaintedFXActor, class AFXActor_X* FXActor);
static class AFXActor_X* PickFXActor(class AFXActor_X* PaintedFXActor, class AFXActor_X*
FXActor, TArray<class UProductAttribute_TA*>& Attributes);
};

// Class TAGame.Facelt_TA
// 0x0160 (0x0060 - 0x01C0)
class UFacelt_TA : public UObject
{
public:
    struct FFaceltMatchInfo           MatchInfo;           // 0x0060 (0x0080)
    [0x0001000000400000] (CPF_NeedCtorLink)
    struct FServerReservationData     ReservationData;     // 0x00E0 (0x0070)
    [0x0001000000400000] (CPF_NeedCtorLink)
    class UAsyncTask*                 JoinTask;           // 0x0150 (0x0008)
    [0x0001000000000000]
    class UFaceltConfig_TA*          Config;              // 0x0158 (0x0008)
    [0x0001800000002000] (CPF_Transient)
    class UOnlineGame_TA*            OnlineGame;         // 0x0160 (0x0008)
    [0x0001800000000000]
    class UOnlineGameParty_X*        OnlineGameParty;    // 0x0168
    (0x0008) [0x0001800000000000]
    class UAsyncTask*                 PendingFindMatchTask; // 0x0170 (0x0008)
    [0x0001000000002000] (CPF_Transient)
    struct FScriptDelegate           __EventJoiningReservation__Delegate; // 0x0178
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __EventAddReservation__Delegate; // 0x0190
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
    struct FScriptDelegate           __EventError__Delegate; // 0x01A8 (0x0018)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.Facelt_TA");
        }
    }

    return uClassPointer;
};

```

```

void __Facelt_TA__Construct_0x1(class UPsyNetService_FaceltMatchComplete_TA* Service);
void __Facelt_TA__FindMatchForPlayer_0x3();
void __Facelt_TA__FindMatchForPlayer_0x2(class UError* Error);
void __Facelt_TA__FindMatchForPlayer_0x1(struct FServerReservationData Reservation, struct FFaceltMatchInfo InMatchInfo, uint64_t Expiration);
void __Facelt_TA__JoinReservation_0x1();
bool IsInOnlineGame();
void OnFailedJoin(class FString FailReason);
void HandleJoinComplete(unsigned long bSuccess, class FString FailReason);
void HandleGameDataSelected(int32_t PlaylistId, int32_t MutatorIndex);
void HandleGRISpawned(class AGRI_X* GRI);
void ClearJoinTask();
void ClearMatchInfo(int32_t InMatchID);
void JoinReservation(struct FServerReservationData Data, struct FFaceltMatchInfo InMatchInfo);
class UErrorType* GetJoinReservationError(struct FFaceltMatchInfo InMatchInfo);
class UAsyncTask* FindMatchForPlayer();
void RemoveMatchNotification(int32_t InMatchID);
void HandleFaceltMatchCancelled(class UPsyNetService_FaceltMatchCancelled_TA* Service);
void eventConstruct();
void EventError(class UError* InError);
void EventAddReservation(struct FServerReservationData InReservationData, struct FFaceltMatchInfo InMatchInfo, uint64_t Expiration, unsigned long bSuppressNotification);
void EventJoiningReservation(class UAsyncTask* Task, struct FFaceltMatchInfo& InMatchInfo);
};

// Class TAGame.FaceltConfig_TA
// 0x0028 (0x0078 - 0x00A0)
class UFaceltConfig_TA : public UOnlineConfig_X
{
public:
    unsigned long          bEnabled : 1;           // 0x0078 (0x0004)
    [0x0001000000000000] [0x00000001]
    TArray<struct FPartnerMapping>      PartnerNames;        // 0x0080
    (0x0010) [0x0001000004000000] (CPF_NeedCtorLink)
    class FString           DefaultPartnerName;     // 0x0090 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FaceltConfig_TA");
        }

        return uClassPointer;
    }

    class FString GetPartnerNameFromID(int32_t Id);
};

// Class TAGame.PsyNetService_FaceltMatchCancelled_TA

```

```
// 0x0004 (0x0090 - 0x0094)
class UPsyNetService_FaceltMatchCancelled_TA : public UPsyNetClientService_X
{
public:
int32_t MatchID; // 0x0090 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_FaceltMatchCancelled_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_FaceltMatchComplete_TA
// 0x0004 (0x0090 - 0x0094)
class UPsyNetService_FaceltMatchComplete_TA : public UPsyNetClientService_X
{
public:
int32_t MatchID; // 0x0090 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_FaceltMatchComplete_TA");
}

return uClassPointer;
};

};

// Class TAGame.FaceltMatchNotification_TA
// 0x00F0 (0x0170 - 0x0260)
class UFaceltMatchNotification_TA : public UNotification_TA
{
public:
struct FServerReservationData ReservationData; // 0x0170 (0x0070)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FFaceltMatchInfo MatchInfo; // 0x01E0 (0x0080)
[0x0001000000400000] (CPF_NeedCtorLink)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FaceltMatchNotification_TA");
}

return uClassPointer;
};

bool ShouldShow();
void ClickAccept();
class UFaceltMatchNotification_TA* SetReservationService(struct FServerReservationData
InReservationData, struct FFaceltMatchInfo InMatchInfo);
};

// Class TAGame.RPC_FaceltFindMatchForPlayer_TA
// 0x0048 (0x00E8 - 0x0130)
class URPC_FaceltFindMatchForPlayer_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;                      // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_FaceltFindMatchForPlayer_TA");
}

return uClassPointer;
};

class UErrorType* eventOverrideErrorType(class UErrorType* ErrorType);
class URPC_FaceltFindMatchForPlayer_TA* SetPlayerID(struct FUniqueNetId Value);
};

// Class TAGame.FaceltFindServerTask_TA
// 0x0028 (0x00D0 - 0x00F8)
class UFaceltFindServerTask_TA : public UAsyncTask
{
public:
float                  SearchTimeout;                // 0x00D0 (0x0004)
[0x0001000000000001] (CPF_Edit)
class URPC_X*           RPC;                      // 0x00D8 (0x0008)
[0x0001000000000000]

```

```

struct FScriptDelegate           __EventResult__Delegate;           // 0x00E0 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FaceltFindServerTask_TA");
}

return uClassPointer;
};

void HandleClientReservationMessage(class UIReservationConnection_X* Connection, class
UFaceltClientReservationMessage_X* Message);
void HandleSearchTimeout();
void HandleJoinMatchError(class URPC_X* InRPC);
void Cleanup();
void Init(class URPC_X* InRPC);
class UFaceltFindServerTask_TA* NotifyOnResult(struct FScriptDelegate Callback);
static class UFaceltFindServerTask_TA* FindUsingRPC(class URPC_X* InRPC);
void EventResult(struct FServerReservationData Reservation, struct FFaceltMatchInfo MatchInfo,
uint64_t Expiration);
};

// Class TAGame.RPC_FaceltPlayerFailedJoin_TA
// 0x0110 (0x00E8 - 0x01F8)
class URPC_FaceltPlayerFailedJoin_TA : public URPC_X
{
public:
int32_t                           MatchID;                      // 0x00E8 (0x0004)
[0x0001000000000000]
int32_t                           FailedJoinServerID;        // 0x00EC (0x0004)
[0x0001000000000000]
class FString                     FailReason;                // 0x00F0 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
struct FServerReservationData     ReservationData;            // 0x0100 (0x0070)
[0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FFaceltMatchInfo          MatchInfo;                 // 0x0170 (0x0080)
[0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
uint64_t                          Expiration;                // 0x01F0 (0x0008)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_FaceltPlayerFailedJoin_TA");
}

```

```

}

return uClassPointer;
};

class UErrorType* eventOverrideErrorHandler(class UErrorType* ErrorType);
class URPC_FaceltPlayerFailedJoin_TA* SetFailReason(class FString InFailReason);
class URPC_FaceltPlayerFailedJoin_TA* SetServerID(int32_t InServerID);
class URPC_FaceltPlayerFailedJoin_TA* SetMatchID(int32_t InMatchID);
};

// Class TAGame.FaceltClientReservationMessage_X
// 0x00FC (0x0060 - 0x015C)
class UFaceltClientReservationMessage_X : public UBeaconMessage_X
{
public:
    struct FServerReservationData             Reservation;           // 0x0060 (0x0070)
    [0x0001000000400000] (CPF_NeedCtorLink)
    struct FFaceltMatchInfo                  MatchInfo;          // 0x00D0 (0x0080)
    [0x0001000000400000] (CPF_NeedCtorLink)
    uint64_t                                Expiration;         // 0x0150 (0x0008)
    [0x0001000000000000]
    int32_t                                 OfflineTimeToLiveSeconds; // 0x0158 (0x0004)
    [0x0001000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FaceltClientReservationMessage_X");
        }
    }

    return uClassPointer;
};

int32_t GetOfflineTimeToLiveSeconds();
class FString GetDSRToken();
class FString GetReservationID();
};

// Class TAGame.GFxData_FaceltMatchNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_FaceltMatchNotification_TA : public UGFxData_Notification_TA
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_FaceltMatchNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.FloppyBits_TA
// 0x001C (0x0074 - 0x0090)
class UFloppyBits_TA : public UAttachmentBehavior_TA
{
public:
class UPhysicsAsset*           PhysAsset;          // 0x0078 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FRBCollisionChannelContainer   CollisionChannels;    // 0x0080
(0x0004) [0x0000000000000000]
class USkeletalMeshComponent*      SkelMesh;          // 0x0088 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FloppyBits_TA");
}

return uClassPointer;
};

void InitPhysics();
void HandleAttached(class UCarMeshComponent_TA* CarMesh);
void OnInit();
};

// Class TAGame.FollowersChangedEvent_TA
// 0x0000 (0x0060 - 0x0060)
class UFollowerChangedEvent_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FollowersChangedEvent_TA");
}

```

```

}

return uClassPointer;
};

};

// Class TAGame.FreeplayCommandsConfig_TA
// 0x0098 (0x0078 - 0x0110)
class UFreeplayCommandsConfig_TA : public UOnlineConfig_X
{
public:
float BallSpinSpeedMultiplier; // 0x0078 (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallInFrontDistance; // 0x007C (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarDistance; // 0x0080 (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarBaseForwardOffset; // 0x0084 (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarForwardOffsetFromTurn; // 0x0088 (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarAccelerationEffect; // 0x008C (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarMaxRightOffset; // 0x0090 (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarSlowRightOffsetPerc; // 0x0094 (0x0004)
[0x0001000000000001] (CPF_Edit)
float BallOnCarMinUpOffsetPerc; // 0x0098 (0x0004)
[0x0001000000000001] (CPF_Edit)
float PopBallUpZVelocity; // 0x009C (0x0004)
[0x0001000000000001] (CPF_Edit)
float MaxVerticalLaunchSpeedRatio; // 0x00A0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float MaxVerticalLaunchSpeedRatioHoops; // 0x00A4 (0x0004)
[0x0001000000000001] (CPF_Edit)
float DefendShotLaunchSpeed; // 0x00A8 (0x0004)
[0x0001000000000001] (CPF_Edit)
float DefendShotLaunchSpeedHoops; // 0x00AC (0x0004)
[0x0001000000000001] (CPF_Edit)
float DefendShotLaunchSpeedBreakout; // 0x00B0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float DefendShotHoopsRecommendedReboundRange; // 0x00B4
(0x0004) [0x0001000000000001] (CPF_Edit)
float DefendShotHoopsSpeedDropoffRange; // 0x00B8 (0x0004)
[0x0001000000000001] (CPF_Edit)
float DefendShotBreakoutSpeedDropoffRange; // 0x00BC
(0x0004) [0x0001000000000001] (CPF_Edit)
float DefendShotBreakoutSpeedDropoffRangeHeight; // 0x00C0
(0x0004) [0x0001000000000001] (CPF_Edit)
float DefendShotHoopsMinSpeedReductionAngle; // 0x00C4
(0x0004) [0x0001000000000001] (CPF_Edit)
float DefendShotHoopsAboveNetSpeedReduction; // 0x00C8
(0x0004) [0x0001000000000001] (CPF_Edit)

```

```

float DefendShotBreakoutMaxTileDistance; // 0x00CC (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassLaunchSpeed; // 0x00D0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassVelocityWeight; // 0x00D4 (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassVelocityZWeight; // 0x00D8 (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassGoalWeight; // 0x00DC (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassMaxGoalLeading; // 0x00E0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassAdditionalHeight; // 0x00E4 (0x0004)
[0x0001000000000001] (CPF_Edit)
float RedirectPassSpeedDropoffRange; // 0x00E8 (0x0004)
[0x0001000000000001] (CPF_Edit)
int32_t RedirectPassIterations; // 0x00EC (0x0004)
[0x0001000000000001] (CPF_Edit)
int32_t DefendShotHoopsMaxSamples; // 0x00F0 (0x0004)
[0x0001000000000001] (CPF_Edit)
int32_t DefendShotHoopsReboundDenominator; // 0x00F4
(0x0004) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* TeleportSucceededSFX; // 0x00F8
(0x0008) [0x0001000000000001] (CPF_Edit)
class UAkSoundCue* TeleportFailedSFX; // 0x0100 (0x0008)
[0x0001000000000001] (CPF_Edit)
class UAkSoundCue* LaunchBallSFX; // 0x0108 (0x0008)
[0x0001000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FreeplayCommandsConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.FreeplayCommandsMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UFreeplayCommandsMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FreeplayCommandsMetrics_TA");
}

return uClassPointer;
};

void CommandsActivatedInSession(TArray<struct FFreeplayCommandEvent>&
ActivatedCommands);
};

// Class TAGame.FreeplayErrors_TA
// 0x0030 (0x0080 - 0x00B0)
class UFreeplayErrors_TA : public UErrorList
{
public:
class UErrorType*           NoCar;                      // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NoBall;                     // 0x0088 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           NoClosestGoal;             // 0x0090 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           TeleportFailed;            // 0x0098 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           InvalidHoopsGoal;          // 0x00A0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           InvalidBreakoutGame;        // 0x00A8 (0x0008)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FreeplayErrors_TA");
}

return uClassPointer;
};

// Class TAGame.FreeplayCommandsStatics_TA
// 0x0000 (0x0060 - 0x0060)
class UFreeplayCommandsStatics_TA : public UObject
{
public:

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FreeplayCommandsStatics_TA");
}

return uClassPointer;
};

static bool IsBallTeleportLocationValid(struct FVector NewLocation, struct FVector CarLocation,
TArray<class UGoal_TA*> Goals);
static struct FReplicatedRBState ModifyRBState(struct FModifyRBStateInParams& InParams);
static float CalculateVerticalLaunchSpeed(struct FCalculateVerticalLaunchSpeedInParams&
InParams);
static struct FVector CalculateLaunchVector(struct FCalculateLaunchVectorInParams&
InParams);
static class UGoal_TA* GetBestGoalFromVelocity(struct FVector InLocation, struct FVector
InVelocity, TArray<class UGoal_TA*> Goals);
static class AActor* GetClosestGoalIndicatorOverride(struct FVector InLocation, class
UGoal_TA* InGoal);
static class UGoal_TA* GetClosestGoal(struct FVector InLocation, TArray<class UGoal_TA*>
Goals);
static float CalculateShotOnGoalSpeedHoops(struct
FCalculateShotOnGoalSpeedHoopsInParams& InParams);
static struct FVector CalculateShotOnGoalLocationHoops(struct
FCalculateShotOnGoalLocationHoopsInParams& InParams);
static struct FVector CalculateShotOnGoalLocationStandard(struct
FCalculateShotOnGoalLocationStandardInParams& InParams);
static struct FVector GetTargetLocationFromFlightTime(struct
FGetTargetLocationFromFlightTimeInParams& InParams);
static float GetFlightTimeToTarget(struct FVector InStartLocation, struct FVector
InTargetLocation, float InHorizontalSpeed);
static struct FVector CalculateRedirectPassLocation(struct
FCalculateRedirectPassLocationInParams& InParams);
static float CalculateRedirectHorizontalLaunchSpeed(struct
FCalculateRedirectHorizontalLaunchSpeedInParams& InParams);
static struct FVector CalculateBallOnCarInAirLocation(struct
FCalculateBallOnCarInAirLocationInParams& InParams);
static struct FVector CalculateBallOnCarLocation(struct FCalculateBallOnCarLocationInParams&
InParams);
static struct FVector CalculateBallInFrontLocation(struct
FCalculateBallInFrontLocationInParams& InParams);
};

// Class TAGame.Goal_Hoops_TA
// 0x0024 (0x01C0 - 0x01E4)
class UGoal_Hoops_TA : public UGoal_TA
{
public:
class AActor* Backboard; // 0x01C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float Radius; // 0x01C8 (0x0004)
};

```

```
[0x0000000000000001] (CPF_Edit)
struct FVector BackboardLocation; // 0x01CC (0x000C)
[0x0000080000000000]
struct FVector BackboardDirection; // 0x01D8 (0x000C)
[0x0000080000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Goal_Hoops_TA");
}

return uClassPointer;
};

float TestSurfaceSDF(struct FVector TestLocation);
void Init();
};

// Class TAGame.FreeplayConfig_TA
// 0x000C (0x0078 - 0x0084)
class UFreeplayConfig_TA : public UOnlineConfig_X
{
public:
float DefaultBoostRechargeDelay; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostRechargeRate; // 0x007C (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bBoostRechargeGroundOnly : 1; // 0x0080 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FreeplayConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.FreeToPlayConfig_TA
// 0x0004 (0x0078 - 0x007C)
class UFreeToPlayConfig_TA : public UOnlineConfig_X
{
```

```
public:  
unsigned long bRequireOnlineSubscription : 1; // 0x0078 (0x0004)  
[0x0000000000000001] [0x00000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.FreeToPlayConfig_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.GFxData_FriendRequestNotification_TA  
// 0x0000 (0x0098 - 0x0098)  
class UGFxData_FriendRequestNotification_TA : public UGFxData_Notification_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_FriendRequestNotification_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.FunctionTemplates  
// 0x0000 (0x0060 - 0x0060)  
class UFunctionTemplates : public UObject  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.FunctionTemplates");  
}
```

```

}

return uClassPointer;
};

static void ArrayFuncs_ShuffleArray_Int(TArray<int32_t>& ArrayToShuffle);
static void Object_SwapArrayItems_Int(int32_t Idx1, int32_t Idx2, TArray<int32_t>& A);
static void Object_Swap_Int(int32_t& A, int32_t& B);
static void Object_SwapArrayItems_TourTeam(int32_t Idx1, int32_t Idx2, TArray<struct FTourTeam>& A);
static void ArrayFuncs_ShuffleArray_MusicTrack_TA(TArray<class UMusicTrack_TA*>& ArrayToShuffle);
static void Object_SwapArrayItems_MusicTrack_TA(int32_t Idx1, int32_t Idx2, TArray<class UMusicTrack_TA*>& A);
static void ArrayFuncs_ShuffleArray_String(TArray<class FString>& ArrayToShuffle);
static void Object_SwapArrayItems_String(int32_t Idx1, int32_t Idx2, TArray<class FString>& A);
static void ArrayFuncs_ShuffleArray_SpecialPickup_TA(TArray<class ASpecialPickup_TA*>& ArrayToShuffle);
static void Object_SwapArrayItems_SpecialPickup_TA(int32_t Idx1, int32_t Idx2, TArray<class ASpecialPickup_TA*>& A);
static class UProductAsset_Skin_TA*
ProductDatabase_TA_TLoadAsset_ProductAsset_Skin_TA(int32_t ProductID);
static void Object_Swap_ClubColorSet(struct FClubColorSet& A, struct FClubColorSet& B);
static void Object_Swap_ClubDetails_X(class UClubDetails_X*& A, class UClubDetails_X*& B);
static class UProductAsset_Body_TA*
ProductDatabase_TA_TLoadAsset_ProductAsset_Body_TA(int32_t ProductID);
};

// Class TAGame.FXActor_Car_Knockout_TA
// 0x001C (0x04894 - 0x04AB0)
class AFXActor_Car_Knockout_TA : public AFXActor_Car_TA
{
public:
TArray<class UFXActorEvent_X*> StateEvents; // 0x04898
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class UFXActorEvent_X* MVPEvent; // 0x049A8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXActor_Car_Knockout_TA");
}

return uClassPointer;
};

void __FXActor_Car_Knockout_TA__HandleReplicatedStateChanged_0x1(class UFXActorEvent_X* E);
void HandleKnockedout(class APRI_KnockOut_TA* InPRI_KO, class ACar_TA* Victim);

```

```

void HandleActiveMVPChanged(class APRI_KnockOut_TA* InPRI_KO);
void HandlePRI(class APRI_KnockOut_TA* InPRI_KO);
void HandleReplicatedStateChanged(class ACar_KnockOut_TA* Car_KO);
void eventDestroyed();
void PostBeginPlay();
};

// Class TAGame.FXActor_TrainingEditor_TA
// 0x0038 (0x0328 - 0x0360)
class AFXActor_TrainingEditor_TA : public AFXActor_X
{
public:
    float AdjustSpeedStateRemoveDelay; // 0x0328 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class UAkSoundCue* AkBallSpeedDirectionChange; // 0x0330
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class UFXActorEvent_X* SimulatingBallState; // 0x0338 (0x0008)
    [0x0000000000000000]
    class UFXActorEvent_X* AdjustingSpeedState; // 0x0340 (0x0008)
    [0x0000000000000000]
    class UFXActorEvent_X* PlaytestingState; // 0x0348 (0x0008)
    [0x0000000000000000]
    class AGameEvent_TrainingEditor_TA* GameEvent; // 0x0350
    (0x0008) [0x0000004000002000] (CPF_Transient)
    float PrevLaunchSpeed; // 0x0358 (0x0004)
    [0x0000004000002000] (CPF_Transient)
    float PrevLaunchSpeedSign; // 0x035C (0x0004)
    [0x0000004000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.FXActor_TrainingEditor_TA");
        }

        return uClassPointer;
    };

    void AdjustSpeedStateEnd();
    void HandleLaunchVelocityChange(class ABall_GameEditor_TA* InBall);
    void HandleBallReset(class ABall_GameEditor_TA* InBall);
    void HandleBallFire(class ABall_GameEditor_TA* InBall);
    void HandlePlaytestStarted(class AGameEvent_TrainingEditor_TA* InGameEvent);
    void HandleGameStateChange(class AGameEvent_TA* InGameEvent);
    void HandleBallRemoved(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* Ball);
    void HandleBallAdded(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* Ball);
    void PostBeginPlay();
};

// Class TAGame.FXExplosionHandler_TA

```

```
// 0x0008 (0x0150 - 0x0158)
class UFXExplosionHandler_TA : public UExplosionHitHandler_TA
{
public:
class AFXActor_X*           FXToApply;          // 0x0150 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXExplosionHandler_TA");
}

return uClassPointer;
};

void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);
};

// Class TAGame.FXTrait_PreviewParticle_TA
// 0x0010 (0x0060 - 0x0070)
class UFXTrait_PreviewParticle_TA : public UFXAttachmentTraitBase_X
{
public:
class UParticleSystem*       ParticlePreview;    // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UParticleSystem*       OldParticle;        // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.FXTrait_PreviewParticle_TA");
}

return uClassPointer;
};

void DisablePreview(class AFXActor_X* __, class UFXActorEvent_X* __);
void EnablePreview(class AFXActor_X* __, class UFXActorEvent_X* __);
void eventInit();
};

// Class TAGame.SeqEvent_StadiumTeamColorsChanged_TA
// 0x0020 (0x017C - 0x019C)
```

```

class USeqEvent_StadiumTeamColorsChanged_TA : public USequenceEvent
{
public:
int32_t Team; // 0x0180 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector Primary; // 0x0184 (0x000C)
[0x0000000000000000]
struct FVector Secondary; // 0x0190 (0x000C)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.SeqEvent_StadiumTeamColorsChanged_TA");
}

return uClassPointer;
};

void LinearColorToVector(struct FLinearColor& InColor, struct FVector& OutVector);
void SetTeamColors(TArray<struct FLinearColor>& Colors);
};

// Class TAGame.GameMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UGameMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameMetrics_TA");
}

return uClassPointer;
};

void StatEvent(struct FUniqueNetId PlayerID, struct FName StatName, class FString HitLocation);
void RecordStatEvent(struct FUniqueNetId PlayerID, struct FName StatName, struct FVector
HitLocation);
void GameEnd(int32_t WinningTeam, int32_t Team0Score, int32_t Team1Score);
void Overtime();
void GoalScoreHit(struct FUniqueNetId PlayerID, class FString CarLocation, class FString
CarVelocity, class FString BallLocation, class FString BallVelocity, float ElapsedTimeSeconds);

```

```

void RecordGoalScoreTouch(struct FBallHitInfo Hit);
void GoalScored(int32_t Team, class FString BallLocation, class FString BallVelocity);
void RecordGoalScored(int32_t Team, struct FVector BallLocation, struct FVector BallVelocity);
void ForfeitVoteFail(int32_t Team);
void ForfeitVotePass(int32_t Team);
void ForfeitVoteStart(struct FUniqueNetId Instigator, int32_t Team);
void LeaveTeam(struct FUniqueNetId PlayerID);
void JoinTeam(struct FUniqueNetId PlayerID, int32_t Team);
void EndRound(int32_t SecondsRemaining);
void StartRound(int32_t SecondsRemaining);
void IdleKick(struct FUniqueNetId PlayerID);
void Disconnect(struct FUniqueNetId PlayerID);
void RemovePlayer(struct FUniqueNetId PlayerID);
void AddPlayer(struct FUniqueNetId PlayerID);
void PlayMutator(struct FName MutatorName);
void GameInit(struct FName MapName, struct FName EventName, struct FName MatchType,
int32_t Playlist, class FString MatchGuid);
};

// Class TAGame.Team_Soccar_TA
// 0x0040 (0x0468 - 0x04A8)
class ATeam_Soccar_TA : public ATeam_TA
{
public:
int32_t GameScore; // 0x0468 (0x0004)
[0x0000000100002020] (CPF_Net | CPF_Transient)
TArray<class AActor*> PrimaryStarts; // 0x0470 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class AActor*> SecondaryStarts; // 0x0480 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FScriptDelegate _EventGameScoreUpdated__Delegate; // 0x0490
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Team_Soccar_TA");
}

return uClassPointer;
};

void OnGameScoreUpdated();
void SetGameScore(int32_t Value);
void ScoreGame();
void InitSpawnPoints();
class AActor* MirrorStartPoint(class AActor* SpawnPoint, struct FVector Pivot);
void eventDestroyed();
void SetGameEvent(class AGameEvent_Team_TA* InGameEvent);
void eventReplicatedEvent(struct FName VarName);

```

```
void EventGameScoreUpdated(class ATeam_Soccar_TA* Team);
};

// Class TAGame.ProductAttribute_ProductLink_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_ProductLink_TA : public UProductAttribute_TA
{
public:
int32_t ProductID; // 0x0080 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_ProductLink_TA");
}

return uClassPointer;
};

bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
};

// Class TAGame.ProductAttribute_Masterpiece_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_Masterpiece_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Masterpiece_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_Level_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_Level_TA : public UProductAttribute_TA
{
public:
int32_t Level; // 0x0080 (0x0004)
```

```
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Level_TA");
}

return uClassPointer;
};

bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
};

// Class TAGame.ProductAttribute_EquipableSlot_TA
// 0x0001 (0x0080 - 0x0081)
class UProductAttribute_EquipableSlot_TA : public UProductAttribute_TA
{
public:
uint8_t Slot; // 0x0080 (0x0001)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_EquipableSlot_TA");
}

return uClassPointer;
};

struct FName GetSlotName();
class FString GetOnlineProductAttributeValue();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
};

// Class TAGame.ProductAttribute_Expiration_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_Expiration_TA : public UProductAttribute_TA
{
public:
uint64_t ExpirationTime; // 0x0080 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Expiration_TA");  
}  
  
return uClassPointer;  
};  
  
class FString eventGetHashID();  
class FString GetTypeHashID();  
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>&  
OtherAttributes);  
};  
  
// Class TAGame.ProductAttribute_Rental_TA  
// 0x0004 (0x0080 - 0x0084)  
class UProductAttribute_Rental_TA : public UProductAttribute_TA  
{  
public:  
int32_t RentalLength; // 0x0080 (0x0004)  
[0x0000004000002000] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Rental_TA");  
}  
  
return uClassPointer;  
};  
  
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>&  
OtherAttributes);  
};  
  
// Class TAGame.ProductAttribute_Indestructible_TA  
// 0x0004 (0x0080 - 0x0084)  
class UProductAttribute_Indestructible_TA : public UProductAttribute_TA  
{  
public:  
unsigned long bIndestructible : 1; // 0x0080 (0x0004)  
[0x0000000000000001] [0x00000001] (CPF_Edit)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;
```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Indestructible_TA");
}

return uClassPointer;
};

bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>&
OtherAttributes);
};

// Class TAGame.ProductAttribute_BlueprintCost_TA
// 0x0004 (0x0080 - 0x0084)
class UProductAttribute_BlueprintCost_TA : public UProductAttribute_TA
{
public:
int32_t Cost; // 0x0080 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_BlueprintCost_TA");
}

return uClassPointer;
};

class FString GetOnlineProductAttributeValue();
bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>&
OtherAttributes);
};

// Class TAGame.ProductAttribute_NoNotify_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_NoNotify_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_NoNotify_TA");
}
}
```

```

return uClassPointer;
};

bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);
};

// Class TAGame.GameEditor_Pawn_TA
// 0x0000 (0x0558 - 0x0558)
class AGameEditor_Pawn_TA : public APawn_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEditor_Pawn_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameEditor_Trigger_TA
// 0x0020 (0x02B0 - 0x02D0)
class AGameEditor_Trigger_TA : public AGameEditor_Actor_TA
{
public:
unsigned long           bEnabled : 1;           // 0x02B0 (0x0004)
[0x0000000000000000] [0x00000001]
struct FScriptDelegate   __OnTriggerTouched__Delegate;    // 0x02B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEditor_Trigger_TA");
}

return uClassPointer;
};

void eventTouch(class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector
HitLocation, struct FVector HitNormal);
void EnableTrigger(unsigned long bEnable);

```

```

void OnTriggerTouched(class AGameEditor_Trigger_TA* Trigger, class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector HitLocation, struct FVector HitNormal);
};

// Class TAGame.GameEditor_Ring_TA
// 0x0030 (0x02D0 - 0x0300)
class AGameEditor_Ring_TA : public AGameEditor_Trigger_TA
{
public:
TArray<struct FActorTouchRecordData> ActorTouchRecords; // 0x02D0
(0x0010) [0x0000000000040000] (CPF_NeedCtorLink)
float MaxRadius; // 0x02E0 (0x0004)
[0x0000000000000000]
uint8_t DetectionType; // 0x02E4 (0x0001)
[0x0000000000000000]
struct FScriptDelegate __OnRingScored__Delegate; // 0x02E8
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEditor_Ring_TA");
}

return uClassPointer;
};

void ModifyBlendedRotation(struct FRotator ControllerRot, struct FRotator DeltaRot, struct FRotator& out_Rotation);
void EnableTrigger(unsigned long bEnable);
struct FRotator GetDesiredRotation();
void Tick(float DeltaTime);
struct FVector GetTouchDirection(class AActor* A);
void OnActorWentThruRing(int32_t RecordIndex);
bool ActorWentThruRing(struct FActorTouchRecordData TouchRecord);
void eventUnTouch(class AActor* Other);
bool AddToBallTouchRecords(struct FActorTouchRecordData NewTouchRecord);
void eventTouch(class AActor* Other, class UPrimitiveComponent* OtherComp, struct FVector HitLocation, struct FVector HitNormal);
bool IsTouchableActor(class AActor* A);
void eventPostBeginPlay();
void OnRingScored(class AGameEditor_Ring_TA* Ring, class AActor* Other);
};

// Class TAGame.GameEditor_Switch_TA
// 0x0000 (0x02D0 - 0x02D0)
class AGameEditor_Switch_TA : public AGameEditor_Trigger_TA
{
public:

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEditor_Switch_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameState_TA
// 0x0010 (0x0060 - 0x0070)
class UGameState_TA : public UObject
{
public:
TArray<struct FComponentTemplate> Components; // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameState_TA");
}

return uClassPointer;
};

};

// Class TAGame.MatchRecorder_TA
// 0x0018 (0x0078 - 0x0090)
class UMatchRecorder_TA : public UMatchRecorder_X
{
public:
TArray<class APRI_TA*> PRIs; // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class AGameEvent_Soccar_TA* SoccarGame; // 0x0088
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.MatchRecorder_TA");
}

return uClassPointer;
};

struct FName __MatchRecorder_TA__SetGameEvent_0x1(class UMutator_TA* Mutator);
void __MatchRecorder_TA__HandleTeamsCreated_0x1(class ATeam_TA* Team);
void UpdatePlayerData(class APRI_TA* PRI);
void Finished();
void HandleMatchEnded(class AGameEvent_Soccar_TA* G);
void HandleMatchStarted(class AGameEvent_TA* G);
void HandleJoinTeam(class ATeam_TA* Team, class APRI_TA* Member);
void HandleTeamsCreated(class AGameEvent_Team_TA* G);
void HandlePlayerRemoved(class AGameEvent_TA* G, class APRI_TA* PRI);
void HandlePlayerAdded(class AGameEvent_TA* G, class APRI_TA* PRI);
void SetClubMatch();
void SetGameEvent(class AGameEvent_TA* InGame);
void AddPlayer(struct FUniqueNetId PlayerID, class FString PlayerName);
};

// Class TAGame.ListenServer_TA
// 0x002C (0x0060 - 0x008C)
class UListenServer_TA : public UObject
{
public:
class AGRI_X* GRI; // 0x0060 (0x0008)
[0x0000800000000000]
class UAdHocBeacon_X* AdHocBeacon; // 0x0068 (0x0008)
[0x0000800000000000]
class AGameEvent_TA* GameEvent; // 0x0070 (0x0008)
[0x0000000000000000]
TArray<struct FLoadingPlayer> LoadingPlayers; // 0x0078 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
unsigned long bRegistered : 1; // 0x0088 (0x0004)
[0x0000004000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ListenServer_TA");
}

return uClassPointer;
};

int32_t __ListenServer_TA__GetNumPlayers_0x1(struct FLoadingPlayer P);
void GoToNextMap();
void HandleCreateLanServerError(class UError* Error);

```

```

void HandleLanServerCreated();
void RegisterLanServer();
void SubmitLanServerMetaData();
class FString BuildLanServerMetaData();
void RebuildReservations();
void TimerUpdateReservations();
void UpdateReservations();
void TimeoutLoadingPlayers();
float GetMapLoadTime(uint8_t Platform);
void RemoveLoadingPlayer(struct FUniqueNetId PlayerID);
int32_t GetMaxPlayers();
int32_t GetNumPlayers();
void KickConnectedPlayers();
struct FUniqueNetId GetCustomMatchOwner();
struct FCustomMatchSettings GetCustomMatchSettings();
void PlayerLoggedOut(class APlayerReplicationInfo* PRI);
void PlayerLoggedIn(class APlayerReplicationInfo* PRI);
bool AllowSplitscreenJoin(struct FUniqueNetId PrimaryPlayerId, struct FUniqueNetId PlayerID,
class FString PlayerName, class FString& Error);
bool IsLocalPlayer(struct FUniqueNetId PlayerID);
bool CanAddPlayers(int32_t Count);
void AllowPlayerLogin(class FString Options, struct FUniqueNetId PlayerID, class FString&
ErrorMessage);
void HandleGRIChanged();
void Init(class AGameEvent_TA* InGameEvent);
};

// Class TAGame.MatchAdminComponent_TA
// 0x0010 (0x0060 - 0x0070)
class UMatchAdminComponent_TA : public UObject
{
public:
class AGameEvent_TA*           GameEvent;          // 0x0060 (0x0008)
[0x0000000000000000]
class UOnlinePlayerPermissions_X* PlayerPermissions; // 0x0068
(0x0008) [0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchAdminComponent_TA");
}

return uClassPointer;
};

bool IsMatchAdmin(class APRI_TA* PRI);
bool IsLocalHost();
void UpdateAll();
void UpdatePlayer(class APRI_TA* PRI);

```

```
void HandlePermissionsUpdated(class UOnlinePlayerPermissions_X* P);
void HandleGameOwnerUpdated(class AGameEvent_TA* G, class APRI_TA* PRI);
void HandlePlayerAdded(class AGameEvent_TA* G, class APRI_TA* PRI);
void Init(class AGameEvent_TA* InGameEvent);
};

// Class TAGame.SpawnDelayConfig_TA
// 0x0004 (0x0078 - 0x007C)
class USpawnDelayConfig_TA : public UOnlineConfig_X
{
public:
unsigned long           bCanDelaySplitscreenPlayerRestart : 1;      // 0x0078
(0x0004) [0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpawnDelayConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_GameEventStateChanged_TA
// 0x0014 (0x017C - 0x0190)
class USeqEvent_GameEventStateChanged_TA : public USequenceEvent
{
public:
class FString           StateName;          // 0x0180 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_GameEventStateChanged_TA");
}

return uClassPointer;
};

};

// Class TAGame.MatchType_Private_TA
// 0x0000 (0x00A8 - 0x00A8)
```

```
class UMatchType_Private_TA : public UMatchType_Custom_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Private_TA");
}

return uClassPointer;
};

void OnInitGameEvent();
bool IsOnlineMultiplayer();
};

// Class TAGame.MatchType_Facelt_TA
// 0x0010 (0x00D0 - 0x00E0)
class UMatchType_Facelt_TA : public UMatchType_Tournament_TA
{
public:
int32_t MinPlayersMetWaitTimeSeconds; // 0x00D0 (0x0004)
[0x0001000000000000]
uint64_t WaitForPlayersExpirationTimeStamp; // 0x00D8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Facelt_TA");
}

return uClassPointer;
};

int32_t GetMaxTeamSize();
bool ShouldShowWaitingForPlayersTime();
bool IsWaitingForPlayersTimeExpired();
bool ShouldCancelMatch();
bool ShouldStartMatch(TArray<struct FUniqueNetId>& OutPlayersAbleToStart);
int32_t GetWaitTimeRemaining();
void OnStartWaitingForPlayers();
};

// Class TAGame.MatchType_Lan_TA
```

```
// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_Lan_TA : public UMatchType_Custom_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_Lan_TA");
}

return uClassPointer;
};

void SetGameData();
};

// Class TAGame.RPC_ChatLog_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_ChatLog_TA : public URPC_X
{
public:
TArray<struct FPlayerChatMessage>          Messages;           // 0x00E8
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
class FString                               GameID;            // 0x00F8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ChatLog_TA");
}

return uClassPointer;
};

class URPC_ChatLog_TA* SetGameID(class FString InGameID);
class URPC_ChatLog_TA* SetMessages(TArray<struct FPlayerChatMessage>& InMessages);
};

// Class TAGame.PlayerSpawnTicket_TA
// 0x0030 (0x0060 - 0x0090)
class UPlayerSpawnTicket_TA : public UObject
{
public:
class AController*             Controller;        // 0x0060 (0x0008)
```

```
[0x0000000000000000]
class APRI_TA*                      PRI;           // 0x0068 (0x0008)
[0x0000000000000000]
float                                SecondsRemaining; // 0x0070 (0x0004)
[0x0000000000000000]
unsigned long                         bWantsRespawn : 1; // 0x0074 (0x0004)
[0x0001000000000000] [0x00000001]
struct FScriptDelegate                __EventRestartPlayer__Delegate; // 0x0078
(0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerSpawnTicket_TA");
}

return uClassPointer;
};

void EventRestartPlayer(class AController* PlayerToRestart);

// Class TAGame.PlayerRemovedEvent_TA
// 0x0008 (0x0060 - 0x0068)
class UPlayerRemovedEvent_TA : public UObject
{
public:
class APRI_TA*                      Player;          // 0x0060 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerRemovedEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameEventFinishedState_TA
// 0x0000 (0x0060 - 0x0060)
class UGameEventFinishedState_TA : public UObject
{
public:
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEventFinishedState_TA");
}

return uClassPointer;
};

};

// Class TAGame.MatchType_FTE_TA
// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_FTE_TA : public UMatchType_Offline_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_FTE_TA");
}

return uClassPointer;
};

bool ShouldAutoSelectTeam(class AController* NewPlayer);
bool ShouldSubmitMatchComplete();
};

// Class TAGame.SpawnPointCluster_TA
// 0x0010 (0x0060 - 0x0070)
class USpawnPointCluster_TA : public UObject
{
public:
TArray<class USpawnPointGroup_TA*>           SpawnTeams;           // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.SpawnPointCluster_TA");
}

return uClassPointer;
};

class USpawnPointGroup_TA* GetGroup(class ATeamInfo* Team);
int32_t GetGroupId(class AActor* SpawnPoint);
void RemoveFromSpawnPoints(class AController* ForPlayer);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct
FRotator& out_Rotation);
bool TryAddToSpawnPointAtIndex(class AController* ForPlayer, int32_t PointIdx);
bool TryAddToSpawnPoint(class AController* ForPlayer);
void SetSpawnPoints(class AGameEvent_TA* GameEvent, struct FVector Center, struct FRotator
Orientation);
void Init(class AGameEvent_TA* GameEvent, struct FVector CenterOffset, struct FRotator
CenterRotation);
};

// Class TAGame.StatCategory_TA
// 0x0014 (0x0060 - 0x0074)
class UStatCategory_TA : public UObject
{
public:
TArray<class UStatTitle_TA*> Titles; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float Multiplier; // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatCategory_TA");
}

return uClassPointer;
};

};

// Class TAGame.ServerPerformanceTracker_TA
// 0x0024 (0x00A4 - 0x00C8)
class UServerPerformanceTracker_TA : public UActorComponent_X
{
public:
class UServerPerformanceConfig_TA* ServerPerformanceConfig; // 0x00A8
(0x0008) [0x0000800000000001] (CPF_Edit)
class UConnectionQualityConfig_TA* ConnectionQualityConfig; // 0x00B0
(0x0008) [0x0000800000000001] (CPF_Edit)
float RollingFpsAverage; // 0x00B8 (0x0004)

```

```

[0x0000000000000000]
float InternalLowFpsTime; // 0x00BC (0x0004)
[0x0000000000000000]
float ServerPerformanceLastUpdateTime; // 0x00C0 (0x0004)
[0x0000000000000000]
unsigned long bReportedServerForPoorPerformance : 1; // 0x00C4
(0x0004) [0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerPerformanceTracker_TA");
}

return uClassPointer;
};

void HandleInternalServerPerformanceCheck(float CurrentFps, float DeltaTime);
uint8_t GetPerformanceStatus(float InPerformance, float MediocreBoundary, float BadBoundary);
void eventTick(float DeltaTime);
void eventAttached();
};

// Class TAGame.MatchBroadcastComponent_TA
// 0x0010 (0x0070 - 0x0080)
class UMatchBroadcastComponent_TA : public UComponent
{
public:
class UMatchBroadcastLog_TA* MatchLog; // 0x0070 (0x0008)
[0x0000000000000000]
class UOnlineGame_Base_X* OnlineGame; // 0x0078 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchBroadcastComponent_TA");
}

return uClassPointer;
};

void AppendBroadcastLog(class AMatchBroadcast_TA* Broadcast, class FString EventJson);
void SetBroadcast(class AMatchBroadcast_TA* Broadcast);
void InitServer(class AGameEvent_Soccar_TA* GameEvent);
void HandleLocalPlayersChanged(class AGameEvent_TA* GameEvent);

```

```

void InitClient(class AGameEvent_Soccar_TA* GameEvent);
bool CreateMatchLog();
class FString GetLogFileName();
void Init(class AGameEvent_Soccar_TA* GameEvent);
};

// Class TAGame.GameEvent_Soccar_SubRules_TA
// 0x0000 (0x0060 - 0x0060)
class UGameEvent_Soccar_SubRules_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Soccar_SubRules_TA");
}

return uClassPointer;
};

void Unapply();
void OnGetBallModifiedScore(class ABall_TA* Ball, int32_t ScoreIndex, int32_t& BallScore);
void OnGetBallSpawnPosition(struct FVector& BallPosition);
bool OnCanUseBallCam();
class ATTeam_TA* OnGetWinningTeam();
bool ShouldSpawnBall();
bool ShouldDisableStatXP();
void SetBallListeners(class ABall_TA* Ball, unsigned long bListen);
void OnRestartPlayer(class AController* NewPlayer);
void OnTick(float DeltaTime);
void OnRemoveGameBall();
void OnAddGameBall(class ABall_TA* Ball);
void OnFirstBallHit();
void OnMatchEnd();
void OnNewRound();
void Init();
};

// Class TAGame.GoalIndicator_TA
// 0x0058 (0x0268 - 0x02C0)
class AGoalIndicator_TA : public AActor
{
public:
class UGoal_TA* Goal; // 0x0268 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class APlayerController_TA* PCOwner; // 0x0270 (0x0008)
[0x0000000000002000] (CPF_Transient)
class APRI_TA* PRI; // 0x0278 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```

class ULensFlareComponent_X*           LensFlareComponent;          // 0x0280
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UStaticMeshComponent*           StaticMeshComponent;        // 0x0288
(0x0008) [0x0000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UStaticMesh*                  DefenseMesh;                // 0x0290 (0x0008)
[0x0000000000000000]
class UStaticMesh*                  OffenseMesh;               // 0x0298 (0x0008)
[0x0000000000000000]
class UMaterialInterface*           MaterialGoalDefense;      // 0x02A0 (0x0008)
[0x0000000000000000]
class UMaterialInterface*           MaterialGoalOffense;       // 0x02A8 (0x0008)
[0x0000000000000000]
TArray<class ULensFlare*>         LensFlareTeamList;        // 0x02B0 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GoalIndicator_TA");
}

return uClassPointer;
};

void HandleViewerTeamChanged(class APRI_X* InPRI);
void HandleViewerReceivedPRI(class APlayerController_X* PC);
void Destroyed();
void SetGoal(class AGameEvent_Soccar_TA* GameEvent, class UGoal_TA* In_Goal);
};

// Class TAGame.StatTitle_TA
// 0x003D (0x0060 - 0x009D)
class UStatTitle_TA : public UObject
{
public:
class FString                   PrimaryTitle;                // 0x0060 (0x0010)
[0x000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                   SecondaryTitle;             // 0x0070 (0x0010)
[0x000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
int32_t                         Points;                    // 0x0080 (0x0004)
[0x0000000000000000]
int32_t                         MinStatValueRequired;    // 0x0084 (0x0004)
[0x0000000000000000]
int32_t                         MaxStatValueAllowed;     // 0x0088 (0x0004)
[0x0000000000000000]
class UStatEvent_TA*            Stat;                      // 0x0090 (0x0008)
[0x0000000000000000]
unsigned long                   bPerStat : 1;            // 0x0098 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long                   bInverseStat : 1;        // 0x0098 (0x0004)

```

```
[0x0000000000000000] [0x00000002]
unsigned long          bClampStatToOne : 1;           // 0x0098 (0x0004)
[0x0000000000000000] [0x00000004]
uint8_t                StatUnitType;             // 0x009C (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatTitle_TA");
}

return uClassPointer;
};

class FString GetUnitTypeAsString(unsigned long bMetric);
bool GetNumPoints(class APRI_TA* Member, class UStatCategory_TA* Category, int32_t& PointsEarned, int32_t& StatCount);
};

// Class TAGame.OnlineGameSkill_TA
// 0x0018 (0x0108 - 0x0120)
class UOnlineGameSkill_TA : public UOnlineGameSkill_X
{
public:
struct FScriptDelegate      __EventPlayerSkillUpdated__Delegate;    // 0x0108
(0x0018) [0x0000000004000000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameSkill_TA");
}

return uClassPointer;
};

void ClientSetSkill(struct FUpdatedPlayerSkillRating Rating, int32_t Playlist);
class APlayerController_TA* GetPlayerController(struct FUniqueNetId PlayerID);
void ReplicatePlayerSkillRating(struct FUniqueNetId PlayerID, int32_t Playlist);
void ReplicateSkillUpdate(class UMatchData_X* MatchData);
void HandleSkillsUpdateFailed(class URPC_X* RPCObject);
void HandleSkillsUpdated(class URPC_UpdateSkills_X* RPCObject);
void EventPlayerSkillUpdated(struct FUniqueNetId PlayerID, struct FUpdatedPlayerSkillRating Rating);
};
```

```

// Class TAGame.PerfMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UPerfMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PerfMetrics_TA");
}

return uClassPointer;
};

void ClientInactiveFPS(class FString MapName, int32_t TotalFrames, int32_t PlaylistId,
TArray<struct FFPSBucketMetrics> Buckets);
void ClientFPS(class FString MapName, int32_t TotalFrames, int32_t PlaylistId, TArray<struct FFPSBucketMetrics> Buckets);
void ServerInactiveFPS(class FString MapName, int32_t TotalFrames, int32_t PlaylistId,
TArray<struct FFPSBucketMetrics> Buckets);
void ServerFPS(class FString MapName, int32_t TotalFrames, int32_t PlaylistId, TArray<struct FFPSBucketMetrics> Buckets);
};

// Class TAGame.RPC_MatchComplete_TA
// 0x0098 (0x00E8 - 0x0180)
class URPC_MatchComplete_TA : public URPC_X
{
public:
int32_t PlaylistId; // 0x00E8 (0x0004)
[0x0000004000000000]
class FString MatchGuid; // 0x00F0 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FOnlinePlayerMatchData> MatchData; // 0x0100
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
struct FGuid AppSessionID; // 0x0110 (0x0010)
[0x0000004000000000]
struct FGuid LevelSessionID; // 0x0120 (0x0010)
[0x0000004000000000]
int32_t MaxTeamSize; // 0x0130 (0x0004)
[0x0000000000000000]
int32_t MatchWinner; // 0x0134 (0x0004)
[0x0000000000000000]
class FString MapName; // 0x0138 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FName MatchType; // 0x0148 (0x0008)
[0x0000000000000000]
TArray<int32_t> TeamScores; // 0x0150 (0x0010)

```

```

[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductDrop>          Drops;                      // 0x0160 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FName>                      StatsToSend;                // 0x0170 (0x0010)
[0x0000000000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_MatchComplete_TA");
}

return uClassPointer;
};

class URPC_MatchComplete_TA* SetGameData(class AGameEvent_Soccar_TA*
GameEvent);
TArray<struct FCurrency> GetCurrencyDrops(struct FUniqueNetId& PlayerID);
TArray<struct FOnlineXPReward> GetRewardDrops(struct FUniqueNetId& PlayerID);
TArray<struct FOnlineProductData> GetDrops(struct FUniqueNetId& PlayerID);
bool ContainsPlayerData(struct FUniqueNetId& PlayerID);
class URPC_MatchComplete_TA* SetLevelSessionID(struct FGuid& InLevelSessionID);
class URPC_MatchComplete_TA* SetAppSessionID(struct FGuid& InAppSessionID);
class URPC_MatchComplete_TA* SetMatchData(TArray<struct FOnlinePlayerMatchData>&
InMatchData);
class URPC_MatchComplete_TA* SetPlaylistID(int32_t InPlaylistID);
};

// Class TAGame.SeqEvent_GameTimeChanged_TA
// 0x000C (0x017C - 0x0188)
class USeqEvent_GameTimeChanged_TA : public USequenceEvent
{
public:
float                  CallTimePercent;           // 0x0180 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long            bHasBeenCalled : 1;      // 0x0184 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_GameTimeChanged_TA");
}

return uClassPointer;
};

```

```
};

// Class TAGame.SeqEvent_GoalScored_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_GoalScored_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_GoalScored_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_ScoreChanged_TA
// 0x000C (0x017C - 0x0188)
class USeqEvent_ScoreChanged_TA : public USequenceEvent
{
public:
int32_t           BlueScore;           // 0x0180 (0x0004)
[0x0000000000000000]
int32_t           RedScore;            // 0x0184 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_ScoreChanged_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_CountDownStart_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_CountDownStart_TA : public USequenceEvent
{
public:
```

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_CountDownStart_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.GameEventMatchActive_TA  
// 0x0000 (0x0060 - 0x0060)  
class UGameEventMatchActive_TA : public UObject  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GameEventMatchActive_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.SeqEvent_CountDownCompleted_TA  
// 0x0004 (0x017C - 0x0180)  
class USeqEvent_CountDownCompleted_TA : public USequenceEvent  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_CountDownCompleted_TA");  
}  
  
return uClassPointer;
```

```
};

};

// Class TAGame.SeqEvent_RoundStart_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_RoundStart_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_RoundStart_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_LiveReplay_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_LiveReplay_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_LiveReplay_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_PodiumStart_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_PodiumStart_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_PodiumStart_TA");
}

return uClassPointer;
};

};

// Class TAGame.Replay_Soccar_TA
// 0x0048 (0x02E0 - 0x0328)
class UReplay_Soccar_TA : public UReplay_TA
{
public:
int32_t TeamSize; // 0x02E0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t UnfairTeamSize; // 0x02E4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long bUnfairBots : 1; // 0x02E8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
int32_t PrimaryPlayerTeam; // 0x02EC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Team0Score; // 0x02F0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Team1Score; // 0x02F4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
TArray<struct FScoredGoal> Goals; // 0x02F8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FHighlight> HighLights; // 0x0308 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FReplayPlayerStats> PlayerStats; // 0x0318 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Replay_Soccar_TA");
}

return uClassPointer;
};

void eventPreExport();
void RemoveTimelineKeyframe(int32_t KeyframeIndex);
void AddTimelineEvent(struct FName EventID, float TimeBeforeEvent);
void RecordUserEvent();

```

```

void RecordHighlight(class APRI_TA* FocusPlayer, struct FName BallName, struct FName
GoalActorName);
void RecordSave(class APRI_TA* Saver, struct FName BallName, struct FName GoalActorName);
void RecordGoal(int32_t TeamNum, class APRI_TA* Scorer, struct FName BallName, struct
FName GoalActorName);
void AddPlayer(class APRI_TA* PRI);
};

// Class TAGame.GameEvent_Football_TA
// 0x0034 (0x0CA0 - 0x0CD4)
class AGameEvent_Football_TA : public AGameEvent_Soccar_TA
{
public:
int32_t LastScoredTeamIndex; // 0x0CA0 (0x0004)
[0x0001000000000000]
struct FVector GoalPlayerSpawnOffset; // 0x0CA4 (0x000C)
[0x0001000000000002] (CPF_Const)
struct FVector BallKickOffDirection; // 0x0CB0 (0x000C)
[0x0001000000000002] (CPF_Const)
struct FVector BallKickOffTorque; // 0x0CBC (0x000C)
[0x0001000000000002] (CPF_Const)
float BallKickOffScale; // 0x0CC8 (0x0004)
[0x0001000000000002] (CPF_Const)
float SpawnRadiusCheck; // 0x0CCC (0x0004)
[0x0001000000000002] (CPF_Const)
float KickoffDelay; // 0x0CD0 (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Football_TA");
}

return uClassPointer;
};

void StartKickoff();
bool __GameEvent_Football_TA__GetBallSpawnGoal_0x1(class UGoal_TA* Goal);
bool ShouldDoKickOff();
int32_t GetGoalScorePoints(class ABall_TA* Ball, class APRI_TA* ScorerPRI, class UGoal_TA*
Goal);
class UGoal_TA* GetBallSpawnGoal();
bool CarFitsAtLocation(struct FVector DesiredLocation);
struct FVector CalculateSpawnLocation(struct FVector StartLocation, struct FRotator
StartRotation, int32_t SpawnLocationIndex);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct
FRotator& out_Rotation);
bool GetBallSpawnOrientation(struct FVector& out_Location, struct FRotator& out_Rotation);
void HandleGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class

```

```

UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
void eventConstruct();
};

// Class TAGame.GameEvent_FTE_TA
// 0x0008 (0x0CA0 - 0x0CA8)
class AGameEvent_FTE_TA : public AGameEvent_Soccar_TA
{
public:
int32_t MaxTimeToShowInstructions; // 0x0CA0 (0x0004)
[0x0001000000000002] (CPF_Const)
int32_t InvalidTeamNum; // 0x0CA4 (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_FTE_TA");
}

return uClassPointer;
};

void __GameEvent_FTE_TA__SubmitMatchComplete_0x1(class APRI_TA* PRI);
void SubmitMatchComplete();
void EndGame();
};

// Class TAGame.Save_TA
// 0x003C (0x0060 - 0x009C)
class USave_TA : public UObject
{
public:
class USaveObjectManager_TA* SaveObjectManager; // 0x0060
(0x0008) [0x0000004000002000] (CPF_Transient)
class FString DirectoryPath; // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString SaveType; // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString SaveExt; // 0x0088 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bExactFileMatch : 1; // 0x0098 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.Save_TA");
}

return uClassPointer;
};

void CancelGetHeaders(struct FScriptDelegate Callback);
void GetHeaders(class ULocalPlayer_TA* Player, class FString FolderPath, class FString Pattern,
struct FScriptDelegate Callback);
class FString GetFullPath(class ULocalPlayer_TA* Player, class FString SaveName);
class FString GetFullSaveDirectory(class ULocalPlayer_TA* Player);
void Delete(class ULocalPlayer_TA* Player, class FString SavePath);
void Save(class ULocalPlayer_TA* Player, class FString SaveName);
void LoadAsync(class ULocalPlayer_TA* Player, class FString SaveFileName, struct
FScriptDelegate Callback);
bool Load(class ULocalPlayer_TA* Player, class FString SaveFileName);
class FString GetUniqueSaveKey(class ULocalPlayer_TA* Player);
void Init();
};

// Class TAGame.SaveData_GameEditor_TA
// 0x0004 (0x009C - 0x00A0)
class USaveData_GameEditor_TA : public USave_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SaveData_GameEditor_TA");
}

return uClassPointer;
};

};

// Class TAGame.IGetRoundNumber_TA
// 0x0000 (0x0060 - 0x0060)
class UIGetRoundNumber_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.IGetRoundNumber_TA");
}

return uClassPointer;
};

int32_t GetRoundNumber();
};

// Class TAGame.GameEvent_GodBall_TA
// 0x0030 (0x0CA0 - 0x0CD0)
class AGameEvent_GodBall_TA : public AGameEvent_Soccar_TA
{
public:
int32_t LastScoredTeamIndex; // 0x0CA0 (0x0004)
[0x0001000000000000]
struct FVector GoalBallSpawnOffset; // 0x0CA4 (0x000C)
[0x0001000000000002] (CPF_Const)
struct FVector GoalPlayerSpawnOffset; // 0x0CB0 (0x000C)
[0x0001000000000002] (CPF_Const)
struct FVector BallKickOffDirection; // 0x0CBC (0x000C)
[0x0001000000000002] (CPF_Const)
float BallKickOffScale; // 0x0CC8 (0x0004)
[0x0001000000000002] (CPF_Const)
float SpawnRadiusCheck; // 0x0CCC (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_GodBall_TA");
}

return uClassPointer;
};

bool __GameEvent_GodBall_TA__GetBallSpawnGoal_0x1(class UGoal_TA* Goal);
class UGoal_TA* GetBallSpawnGoal();
bool CarFitsAtLocation(struct FVector DesiredLocation);
struct FVector CalculateSpawnLocation(struct FVector StartLocation, struct FRotator StartRotation, int32_t SpawnLocationIndex);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct FRotator& out_Rotation);
bool GetBallSpawnOrientation(struct FVector& out_Location, struct FRotator& out_Rotation);
void HandleGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
void eventConstruct();
};

```

```

// Class TAGame.StatFactory_KnockOut_TA
// 0x00C0 (0x0458 - 0x0518)
class AStatFactory_KnockOut_TA : public AStatFactory_TA
{
public:
struct FStatEventCollection_KO           Events_KO;           // 0x0458 (0x0098)
[0x0001000000000002] (CPF_Const)
class AGameEvent_KnockOut_TA*           KnockOutGame;       // 0x04F0
(0x0008) [0x000100000002000] (CPF_Transient)
float                                AerialHitFloorDistance; // 0x04F8 (0x0004)
[0x0001000000000002] (CPF_Const)
struct FScriptDelegate                __EventPlayerKnockedOut__Delegate; // 0x0500
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatFactory_KnockOut_TA");
}

return uClassPointer;
};

void OnCarDemolished(class ACar_TA* Victim, struct FDemolishData Data);
void HandleMatchMVP(class APRI_TA* PRI);
void OnMatchEnded(class AGameEvent_Soccar_TA* GameEvent);
void HandlePlayerKnockedOut(class APRI_KnockOut_TA* PRI, class ACar_TA* Victim);
void HandleThrown(class ACar_KnockOut_TA* Car_KO, class ACar_KnockOut_TA* Thrower);
void HandleGrabbed(class ACar_KnockOut_TA* Car_KO, class ACar_KnockOut_TA* Grabber);
void HandleBlockTaken(class ACar_KnockOut_TA* Car_KO, TArray<class ACar_KnockOut_TA*> Blockers, unsigned long bHeavy);
void HandleDamageTaken(class ACar_KnockOut_TA* Car_KO, TArray<class ACar_KnockOut_TA*> Attackers, int32_t Amount, unsigned long bHeavy);
void UnregisterCar(class APawn_X* Car);
void RegisterCar(class ACar_TA* Car);
void SetGameEvent(class AGameEvent_Soccar_TA* InGameEvent);
void EventPlayerKnockedOut(class ACar_KnockOut_TA* Victim);
};

// Class TAGame.VehiclePickup_Item_TA
// 0x0018 (0x02F0 - 0x0308)
class AVehiclePickup_Item_TA : public AVehiclePickup_TA
{
public:
TArray<class ASpecialPickup_TA*>      ItemArchetypes;      // 0x02F0
(0x0010) [0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class AFXActor_X*                      ReplicatedFXActorArchetype; // 0x0300
(0x0008) [0x0000004100000020] (CPF_Net)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VehiclePickup_Item_TA");
}

return uClassPointer;
};

void Pickup(class ACar_TA* Car);
bool CanPickup(class ACar_TA* Car);
void SetFXActorArchetype(class AFXActor_X* InArchetype);
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.PlayerStart_Platform_TA
// 0x0010 (0x0398 - 0x03A8)
class APlayerStart_Platform_TA : public APlayerStart
{
public:
class UStaticMeshComponent*           StaticMeshComponent;          // 0x0398
(0x0008) [0x00010000040A000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_EditConst |
CPF_Component | CPF_EditInline)
float                                HideTime;                  // 0x03A0 (0x0004)
[0x0001000000000003] (CPF_Edit | CPF_Const)
unsigned long                         bActive : 1;            // 0x03A4 (0x0004)
[0x0001000100002020] [0x00000001] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerStart_Platform_TA");
}

return uClassPointer;
};

void RelnitPhysics();
void hide();
void TryToStartHideTimer(class AGameEvent_TA* GameEvent);
void HandleGameStateChanged(class AGameEvent_TA* GameEvent);
void Show();
void OnActiveChanged();
void eventPostBeginPlay();
void eventReplicatedEvent(struct FName VarName);
};

```

```
// Class TAGame.GameEvent_Lobby_TA
// 0x0010 (0x0738 - 0x0748)
class AGameEvent_Lobby_TA : public AGameEvent_TA
{
public:
class ABall_TA* BallArchetype; // 0x0738 (0x0008)
[0x0000000000000001] (CPF_Edit)
class ABall_TA* Ball; // 0x0740 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Lobby_TA");
}

return uClassPointer;
};

void SpawnBall();
bool ShouldShowDisconnectedPlayersOnScoreboard();
bool AllowShutdown();
void eventDestroyed();
bool CarFitsAtLocation(struct FVector DesiredLocation, float RadiusCheck);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct FRotator& out_Rotation);
void HandleBallAtGoal(class ABall_TA* ParticularBall, class UGoal_TA* Goal);
void OnCarSpawned(class ACar_TA* NewCar);
void InitField();
void eventPostBeginPlay();
void StartFirstState();
void OnInit();
};

// Class TAGame.GameEvent_PostGameLobby_TA
// 0x0000 (0x0748 - 0x0748)
class AGameEvent_PostGameLobby_TA : public AGameEvent_Lobby_TA
{
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_PostGameLobby_TA");
}
}
```

```

return uClassPointer;
};

bool ShouldShowBallIndicator();
bool IsPostMatch();
void StartFirstState();
};

// Class TAGame.SimilarSeasonLogos_TA
// 0x0010 (0x0060 - 0x0070)
class USimilarSeasonLogos_TA : public UObject
{
public:
TArray<struct FSimilarLogoGroup> LogoGroups; // 0x0060
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SimilarSeasonLogos_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameEvent_Soccar_SubRules_Basketball_TA
// 0x0008 (0x0060 - 0x0068)
class UGameEvent_Soccar_SubRules_Basketball_TA : public UGameEvent_Soccar_SubRules_TA
{
public:
int32_t LongDistanceScore; // 0x0060 (0x0004)
[0x0000000000000000]
int32_t DefaultScore; // 0x0064 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GameEvent_Soccar_SubRules_Basketball_TA");
}

return uClassPointer;
};

```

```

void OnGetBallModifiedScore(class ABall_TA* Ball, int32_t ScoreIndex, int32_t& BallScore);
};

// Class TAGame.GameEvent_Soccar_SubRules_Items_TA
// 0x0028 (0x0060 - 0x0088)
class UGameEvent_Soccar_SubRules_Items_TA : public UGameEvent_Soccar_SubRules_TA
{
public:
    class UPlayerItemDispenser_TA* DispenserArchetype; // 0x0060
    (0x0008) [0x0000000000000001] (CPF_Edit)
    float StaggerTime; // 0x0068 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float CurrentStaggerByTeam[0x2]; // 0x006C (0x0008)
    [0x0000000000002000] (CPF_Transient)
    TArray<class UPlayerItemDispenser_TA*> ItemDispensers; // 0x0078
    (0x0010) [0x000008000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Soccar_SubRules_Items_TA");
        }

        return uClassPointer;
    }

    void RandomizeDispenserOrder();
    void CheckPlayers();
    void Unapply();
    void UnregisterEvents();
    void OnMatchEnd();
    void HandleGoalScored(class AGameEvent_Soccar_TA* GameEvent, class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIdx);
    void HandleCarSpawned(class AGameEvent_TA* GameEvent, class ACar_TA* Car);
    void HandleStartNewRound(class AGameEvent_Soccar_TA* GameEvent);
    void HandleTeamsCreated(class AGameEvent_Team_TA* GameEvent);
    void HandlePRIRemoved(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
    void HandlePRIAdded(class AGameEvent_TA* GameEvent, class APRI_TA* PRI);
    void OnNewRound();
    void StopUpdating();
    void Init();
};

// Class TAGame.PlayerItemDispenser_TA
// 0x0028 (0x0060 - 0x0088)
class UPlayerItemDispenser_TA : public UObject
{
public:
    float ItemGiveRate; // 0x0060 (0x0004)
}

```

```

[0x0000000000000001] (CPF_Edit)
float PreviewTimeSeconds; // 0x0064 (0x0004)
[0x0000080000000001] (CPF_Edit)
class UItemPool_TA* ItemPool; // 0x0068 (0x0008)
[0x000000004400009] (CPF_Edit | CPF_ExportObject | CPF_NeedCtorLink | CPF_EditInline)
unsigned long bResetTimerOnDemolish : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bWaitingForNextRound : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bGiveItemOnRoundStart : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long bGiveItemOnFirstBallHit : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
unsigned long bGiveItemBeforeRoundStart : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000010] (CPF_Edit)
unsigned long bHideItemIconByDefault : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000020] (CPF_Edit)
unsigned long bGiven : 1; // 0x0070 (0x0004)
[0x0000000000002000] [0x00000040] (CPF_Transient)
float NextItemTime; // 0x0074 (0x0004)
[0x0000000000002000] (CPF_Transient)
float NextItemPreviewTime; // 0x0078 (0x0004)
[0x0000000000002000] (CPF_Transient)
class APRI_TA* PRI; // 0x0080 (0x0008)
[0x0000080000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerItemDispenser_TA");
}

return uClassPointer;
};

void GiveNextItemNextUpdate();
void ClearNextItemPreviewTime();
void ClearNextItemTime();
void SetNextItemTime(float TimeFromNow);
void OnGoalScored();
void OnMatchEnd();
void OnNewRound(float InStaggerTime, float InGameTimeRemaining);
void Update();
float GetTotalGameTimePlayed();
float GetCurrentPickupCooldownTime();
void SetCurrentPickupCooldownTime(float NewCooldownSeconds);
void SetPickupCooldownTime(float NewCooldownSeconds);
void HandleActiveRoundChanged(class AGameEvent_Soccar_TA* InGameEvent);
void HandleFirstBallHit(class AGameEvent_Soccar_TA* InGameEvent);
void HandleCarPickupsSet();

```

```

void HandleCarDestroyed(class APawn_X* Car);
void HandleCarDemolished(class ACar_TA* Car, struct FDemolishData Data);
void HandleCarSet(class APRI_TA* OwningPRI);
void SetPreviewTime(float NewTimeSeconds);
void Init(class APRI_TA* InPRI, float InStaggerTime);
};

// Class TAGame.GameEvent_Training_Goalie_TA
// 0x0004 (0xEF8 - 0xEFC)
class AGameEvent_Training_Goalie_TA : public AGameEvent_Training_TA
{
public:
int32_t Score; // 0xEF8 (0x0004)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Training_Goalie_TA");
}

return uClassPointer;
};

void CommitRedoRound();
void ResetGameEvent();
void GetBallAimInfo(struct FVector BallSpawnLoc, struct FFieldSpawnInfo SpawnInfo, struct FVector& Out_GoalAimLocation, float& Out_DepthPercent, float& Out_HeightPercent, float& Out_WidthPercent);
struct FVector GetBallArchVelocity(struct FVector BallSpawnLoc, struct FFieldSpawnInfo SpawnInfo);
struct FVector GetBallForwardVelocity(struct FVector BallSpawnLoc, struct FFieldSpawnInfo SpawnInfo);
struct FVector GetBallAimVelocity(struct FVector BallSpawnLoc, struct FFieldSpawnInfo SpawnInfo);
struct FVector DetermineBallInitialVelocity(struct FVector BallSpawnLoc, class ACar_TA* Car, struct FFieldSpawnInfo SpawnInfo);
struct FRotator GetCarSpawnRotation(struct FFieldSpawnInfo SpawnInfo, struct FVector CarSpawnLocation);
struct FVector GetCarSpawnLocation(struct FFieldSpawnInfo SpawnInfo);
int32_t GetScore();
};

// Class TAGame.GameEvent_Training_Aerial_TA
// 0x0000 (0xEF8 - 0xEF8)
class AGameEvent_Training_Aerial_TA : public AGameEvent_Training_TA
{
public:

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Training_Aerial_TA");
}

return uClassPointer;
};

void CheckForReset();
struct FVector DetermineBallInitialVelocity(struct FVector BallSpawnLoc, class ACar_TA* Car,
struct FFieldSpawnInfo SpawnInfo);
};

// Class TAGame.GameEvent_Training_Striker_TA
// 0x0000 (0x0EF8 - 0x0EF8)
class AGameEvent_Training_Striker_TA : public AGameEvent_Training_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameEvent_Training_Striker_TA");
}

return uClassPointer;
};

void CheckForReset();
struct FVector DetermineBallInitialVelocity(struct FVector BallSpawnLoc, class ACar_TA* Car,
struct FFieldSpawnInfo SpawnInfo);
int32_t GetScore();
};

// Class TAGame.TutorialSave_TA
// 0x0030 (0x00C8 - 0x00F8)
class UTutorialSave_TA : public USaveObject_TA
{
public:
TArray<class FString> TrainingEventsPlayed; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class FString> TrainingEventsPerfected; // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> DetailsGroupsSeen; // 0x00E8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TutorialSave_TA");
}

return uClassPointer;
};

bool HasSeenDetailsGroup(int32_t DetailsGroup);
void MarkDetailsGroupSeen(int32_t DetailsGroup);
bool AddToTrainingPerfectList_Internal(class FString EventName);
bool AddToTrainingPerfectList(class FString EventName, class FString Difficulty);
bool HasPerfectedTraining_Internal(class FString EventName);
bool HasPerfectedTraining(class FString EventName, class FString Difficulty);
bool AddToTutorialPlayedList_Internal(class FString EventName);
bool AddToTutorialPlayedList(class FString EventName, class FString Difficulty);
bool HasPlayedTraining_Internal(class FString EventName);
bool HasPlayedTraining(class FString EventName, class FString Difficulty);
};

// Class TAGame.UUIStrings_TA
// 0x0180 (0x0060 - 0x01E0)
class UUIStrings_TA : public UUIStrings_X
{
public:
class FString Continue; // 0x0060 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString ExitGame; // 0x0070 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Retry; // 0x0080 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString OK; // 0x0090 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Cancel; // 0x00A0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Yes; // 0x00B0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString No; // 0x00C0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString SignIn; // 0x00D0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Days; // 0x00E0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Hours; // 0x00F0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Minutes; // 0x0100 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Seconds; // 0x0110 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

```

```

class FString Meters; // 0x0120 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Kilometers; // 0x0130 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Feet; // 0x0140 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Miles; // 0x0150 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString MetersUnit; // 0x0160 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString FeetUnit; // 0x0170 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString MetersPerSecondUnit; // 0x0180 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString FeetPerSecondUnit; // 0x0190 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Error; // 0x01A0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Warning; // 0x01B0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Close; // 0x01C0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString Savage; // 0x01D0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UIStrings_TA");
}

return uClassPointer;
};

};

// Class TAGame.TrainingEditorMirror_TA
// 0x0044 (0x0060 - 0x00A4)
class UTrainingEditorMirror_TA : public UObject
{
public:
class AGameEvent_TrainingEditor_TA* TrainingEditor; // 0x0060
(0x0008) [0x00100000000000]
int32_t CurrentRoundNumber; // 0x0068 (0x0004)
[0x00100000002000] (CPF_Transient)
struct FVector MirrorPoint; // 0x006C (0x000C)
[0x00100000002000] (CPF_Transient)
struct FVector MirrorLocationNormal; // 0x0078 (0x000C)
[0x00100000002000] (CPF_Transient)
struct FRotator MirrorRotationAngle; // 0x0084 (0x000C)

```

```

[0x0001000000002000] (CPF_Transient)
class APylon_Soccar_TA* Pylon; // 0x0090 (0x0008)
[0x0001800000002000] (CPF_Transient)
class AMirrorAxis_TA* MirrorOverride; // 0x0098 (0x0008)
[0x0001800000002000] (CPF_Transient)
unsigned long bMirrorShot : 1; // 0x00A0 (0x0004)
[0x0001000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingEditorMirror_TA");
}

return uClassPointer;
};

void __TrainingEditorMirror_TA__MirrorCurrentShot_0x2(class ACar_TA* Car);
void __TrainingEditorMirror_TA__MirrorCurrentShot_0x1(class ABall_GameEditor_TA* Ball);
void CacheMirrorInfoFromMirrorAxis(class AMirrorAxis_TA* InMirrorAxis);
void CacheMirrorInfoFromPylonAndMapInfo(class APylon_Soccar_TA* InPylon, class UMapInfo_TA* InMapInfo);
void ClearMirrorCache();
void MirrorCurrentShot();
class UMapInfo_TA* GetMapInfo();
void HandleMirrorAxisInjected();
void HandlePylonInjected();
void HandleRoundStarted(class AGameEvent_TrainingEditor_TA* GameEvent);
void HandleGameEventCommand(class APlayerController* Caller, class FString CommandString);
void InitTrainingEditor(class AGameEvent_TrainingEditor_TA* InTrainingEditor);
};

// Class TAGame.TrainingEditorManipulationConfig_TA
// 0x0004 (0x0078 - 0x007C)
class UTrainingEditorManipulationConfig_TA : public UOnlineConfig_X
{
public:
unsigned long bEnableMirroring : 1; // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingEditorManipulationConfig_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.TrainingEditorNavigationConfig_TA
// 0x0004 (0x0078 - 0x007C)
class UTrainingEditorNavigationConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bEnablePlaylistOverride : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long          bEnableShuffling : 1;           // 0x0078 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingEditorNavigationConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.SaveData_GameEditor_Training_TA
// 0x0058 (0x00A0 - 0x00F8)
class USaveData_GameEditor_Training_TA : public USaveData_GameEditor_TA
{
public:
class FString          LoadedSaveName;           // 0x00A0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTrainingEditorData_TA*   TrainingData;           // 0x00B0 (0x0008)
[0x0000000000000000]
int32_t                PlayerTeamNumber;           // 0x00B8 (0x0004)
[0x0000000000000000]
unsigned long          bUnowned : 1;           // 0x00BC (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bPerfectCompleted : 1;           // 0x00BC (0x0004)
[0x0000000000000000] [0x00000002]
int32_t                ShotsCompleted;           // 0x00C0 (0x0004)
[0x0000000000000000]
class FString          FavoritesFolderPath;           // 0x00C8 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString          MyTrainingFolderPath;           // 0x00D8 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString          DownloadedFolderPath;           // 0x00E8 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SaveData_GameEditor_Training_TA");
}

return uClassPointer;
};

class FString GetTrainingFullPath(class ULocalPlayer_TA* Player, uint8_t InSaveType);
class FString GetTrainingPath(uint8_t InSaveType);
class FString GetTrainingFolder(uint8_t InSaveType);
static uint8_t GetTrainingSaveType(unsigned long bOwned, unsigned long bFavorited);
class FString GetFullSavePath(class ULocalPlayer_TA* Player, class FString SaveName);
class FString GetFullSaveDirectory(class ULocalPlayer_TA* Player);
void Init();
};

// Class TAGame.TrainingEditorMetrics_TA
// 0x002C (0x0080 - 0x00AC)
class UTrainingEditorMetrics_TA : public UMetricsGroup_X
{
public:
class UTrainingMetricsConfig_TA*           TrainingMetricsConfig;          // 0x0080
(0x0008) [0x0000800000000000]
TArray<struct FTrainingNavigationEvent>   NavigationCommands;          // 0x0088
(0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FTrainingManipulationEvent>   ManipulationCommands;        //
0x0098 (0x0010) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                                     BatchThreshold;              // 0x00A8 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingEditorMetrics_TA");
}

return uClassPointer;
};

void TrainingCommandsUsed(struct FGuid Id, class FString PackCode, TArray<struct FTrainingNavigationEvent>& NavigationEvents, TArray<struct FTrainingManipulationEvent>& ManipulationEvents);
void ClearUsedCommands();

```

```

void SendAndClearUsedCommands(struct FGuid Id, class FString PackCode);
void SendUsedCommandsBatchIfFull(struct FGuid Id, class FString PackCode);
void TrainingManipulationCommand(struct FTrainingManipulationEvent InEvent);
void TrainingNavigationCommand(struct FTrainingNavigationEvent InEvent);
void TrainingProgressReset(struct FGuid Id, class FString PackCode);
void TrainingShotComplete(struct FGuid Id, class FString PackCode, int32_t Index, unsigned long bSuccess, int32_t ShotAttempt);
void TrainingShotAttempt(struct FGuid Id, class FString PackCode, int32_t Index, int32_t ShotAttempt);
void TrainingSetComplete(struct FGuid Id, class FString PackCode);
void TrainingSetExit(struct FGuid Id, class FString PackCode);
void TrainingSetEnter(struct FGuid Id, class FString PackCode);
void TrainingSetDelete(struct FGuid Id, class FString PackCode);
void TrainingSetDownload(struct FGuid Id, class FString PackCode);
void TrainingEditorExit(struct FGuid Id, class FString PackCode);
void TrainingEditorEnter(struct FGuid Id, class FString PackCode);
void TrainingSetUpdated(struct FGuid Id, class FString PackCode, class FString TrainingType, int32_t Difficulty, int32_t TotalRounds, class FString UserTags);
void TrainingSetSubmitted(struct FGuid Id, class FString PackCode, class FString TrainingType, int32_t Difficulty, int32_t TotalRounds, class FString UserTags);
void TrainingSetCreate(struct FGuid Id, class FString PackCode, class FString TrainingType, int32_t Difficulty, class FString UserTags, unsigned long bSuccessfullyCreated);
};

// Class TAGame.TrainingEditorData_TA
// 0x00E0 (0x0060 - 0x0140)
class UTrainingEditorData_TA : public UObject
{
public:
    struct FGuid           TM_Guid;           // 0x0060 (0x0010)
    [0x0000000000000000]
    class FString          Code;              // 0x0070 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString          TM_Name;           // 0x0080 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    uint8_t                Type;              // 0x0090 (0x0001)
    [0x0000000040000000] (CPF>EditInlineNotify)
    uint8_t                Difficulty;        // 0x0091 (0x0001)
    [0x0000000040000000] (CPF>EditInlineNotify)
    class FString          CreatorName;       // 0x0098 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString          Description;        // 0x00A8 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    TArray<int32_t>        Tags;              // 0x00B8 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FName            MapName;           // 0x00C8 (0x0008)
    [0x0000000040000000] (CPF>EditInlineNotify)
    int32_t                NumRounds;         // 0x00D0 (0x0004)
    [0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
    uint64_t                CreatedAt;          // 0x00D8 (0x0008)
    [0x0000000040000000] (CPF>EditInlineNotify)
    uint64_t                UpdatedAt;          // 0x00E0 (0x0008)
    [0x0000000040000000] (CPF>EditInlineNotify)
    struct FUniqueNetId     CreatorPlayerID;   // 0x00E8 (0x0048)
};

```

```
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FEditorRoundData>          Rounds;                                // 0x0130 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingEditorData_TA");
}

return uClassPointer;
};

void Init();
};

// Class TAGame.TrainingPackProgress_TA
// 0x0058 (0x0060 - 0x00B8)
class UTrainingPackProgress_TA : public UObject
{
public:
class FString           PackCode;          // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FTrainingRoundProgress>    Progress;          // 0x0070
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
uint64_t                  TimeLastPlayed; // 0x0080 (0x0008)
[0x0000000000000000]

struct FScriptDelegate      _EventProgressUpdated__Delegate; // 0x0088
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      _ValidateRound__Delegate;     // 0x00A0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingPackProgress_TA");
}

return uClassPointer;
};

bool SetAttemptByRoundNumber(int32_t RoundNumber, uint8_t NewAttempt, unsigned long bForceSet);
void ClearUncompletedShots();
void InitProgress(class UTrainingEditorData_TA* TrainingData, struct FScriptDelegate ValidateRoundCallback);
```

```
bool ValidateRound(struct FEditorRoundData RoundData);
void EventProgressUpdated(class UTrainingPackProgress_TA* NewProgress);
};

// Class TAGame.TrainingProgressSave_TA
// 0x0028 (0x00C8 - 0x00F0)
class UTrainingProgressSave_TA : public USaveObject_TA
{
public:
TArray<class UTrainingPackProgress_TA*> AllPacksProgress; // 0x00C8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __ValidateRound__Delegate; // 0x00D8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingProgressSave_TA");
}

return uClassPointer;
};

class UTrainingPackProgress_TA* GetPackProgress(class UTrainingEditorData_TA*
TrainingData, struct FScriptDelegate ValidateRoundCallback);
bool ValidateRound(struct FEditorRoundData RoundData);
};

// Class TAGame.IStartRound_TA
// 0x0000 (0x0060 - 0x0060)
class UIStartRound_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IStartRound_TA");
}

return uClassPointer;
};

void StartAtRoundNumber(int32_t NewRoundNumber);
};
```

```

// Class TAGame.ProductAssetCache_TA
// 0x0038 (0x0070 - 0x00A8)
class UProductAssetCache_TA : public UComponent
{
public:
unsigned long          bDebug : 1;                                // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
TArray<int32_t>        NewProductIDs;                           // 0x0078 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>        LoadedProductIDs;                         // 0x0088 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UProductAsset_TA*> Assets;                      // 0x0098 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAssetCache_TA");
}

return uClassPointer;
};

class FString GetDebugName();
void HandleProductLoaded(struct FAssetLoadResult Result);
void FinishedAddingProducts();
void AddProducts(TArray<int32_t> InProductIDs);
void Clear();
};

// Class TAGame.MapSelector_TA
// 0x0098 (0x0070 - 0x0108)
class UMapSelector_TA : public UComponent
{
public:
unsigned long          bDebug : 1;                                // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
TArray<class UMapData_TA*> AllMaps;                            // 0x0078 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UMapData_TA*> FilteredMaps;                         // 0x0088 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FMapGroup> AllGroups;                             // 0x0098 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FMapGroup> RemainingGroupMaps;                   // 0x00A8
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>        RemainingGroups;                          // 0x00B8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<int32_t>        PreviousGroups;                          // 0x00C8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t                PreviousGroupMax;                        // 0x00D8 (0x0004)

```

```

[0x0000000000000001] (CPF_Edit)
class UMapData_TA* LastPlayedMap; // 0x00E0 (0x0008)
[0x000004000002000] (CPF_Transient)
float Skill; // 0x00E8 (0x0004)
[0x000004000002000] (CPF_Transient)
TArray<struct FMapSkillSettings> MapSkillRequirements; // 0x00F0
(0x0010) [0x000004000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FName MapSetName; // 0x0100 (0x0008)
[0x00000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapSelector_TA");
}

return uClassPointer;
};

struct FMapGroup __MapSelector_TA__GetNextMap_0x1(int32_t GroupIdx);
struct FName GetNextPreferredMapName(int32_t NumPlayers, TArray<struct FName>& Likes,
TArray<struct FName>& Dislikes);
class UMapData_TA* GetNextPreferredMap(int32_t NumPlayers, TArray<struct FName>& Likes,
TArray<struct FName>& Dislikes);
struct FMapGroup GetChosenGroup(TArray<float> GroupWeights);
TArray<float> GetMapGroupWeights(int32_t NumPlayers, TArray<struct FName>& Likes,
TArray<struct FName>& Dislikes);
float GetMapGroupWeight(struct FMapGroup Group, int32_t NumPlayers, TArray<struct
FName>& Likes, TArray<struct FName>& Dislikes);
struct FName GetNextMapName();
class FString GetMapsString(TArray<class UMapData_TA*> Maps);
void SetMapPlayed(class UMapData_TA* Map);
class UMapData_TA* GetNextMap();
void RebuildRemainingGroups();
void FilterRemainingGroupsBySkill();
TArray<class UMapData_TA*> FilterMapsBySkill(TArray<class UMapData_TA*> InMaps);
void RebuildMaps();
void SetSkill(float InSkill, TArray<struct FMapSkillSettings> InSkillReqs);
void SetMaps(TArray<class UMapData_TA*> Maps);
};

// Class TAGame.GameInfo_Cinematic_TA
// 0x0000 (0x04D8 - 0x04D8)
class AGameInfo_Cinematic_TA : public AGameInfoBase_TA
{
public:

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Cinematic_TA");
}

return uClassPointer;
};

void SetStadiumColors();
void GenericPlayerInitialization(class AController* C);
class APlayerController* SpawnPlayerController(struct FVector SpawnLocation, struct FRotator
SpawnRotation);
void InitGame(class FString Options, class FString& ErrorMessage);
};

// Class TAGame.PlayerController_Cinematic_TA
// 0x0008 (0x0878 - 0x0880)
class APlayerController_Cinematic_TA : public APlayerController_X
{
public:
class UPostProcessManager_X*           PostProcessManagerArchetype;      // 0x0878 (0x0008) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerController_Cinematic_TA");
}

return uClassPointer;
};

void eventReceivedPlayer();

// Class TAGame.GameInfo_CollisionTest_TA
// 0x002C (0x04D8 - 0x0504)
class AGameInfo_CollisionTest_TA : public AGameInfoBase_TA
{
public:
TArray<class UCollisionTestComponent_TA*>   TestComponents;          // 0x04D8 (0x0010) [0x000000004482008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_NeedCtorLink | CPF_EditInline)
class UCollisionTestComponent_TA*           CurrentTest;            // 0x04E8 (0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
int32_t           CurrentIndex;           // 0x04F0 (0x0004) [0x0000000000002000] (CPF_Transient)

```

```
class AActor* CurrentActor; // 0x04F8 (0x0008)
[0x0000000000002000] (CPF_Transient)
unsigned long bRepeat : 1; // 0x0500 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_CollisionTest_TA");
}

return uClassPointer;
};

class AActor* SpawnActor(class UCollisionTestComponent_TA* Test, float Alpha);
class AActor* SpawnActorFromTest(class UCollisionTestComponent_TA* Test);
void SkipTest();
void Repeat(int32_t Index);
void SkipTo(int32_t Index);
void Skip(int32_t Steps);
void FireNext();
void StartNextTest();
void FindTestActors();
void SetStadiumColors();
void GenericPlayerInitialization(class AController* C);
class APlayerController* SpawnPlayerController(struct FVector SpawnLocation, struct FRotator
SpawnRotation);
void InitGame(class FString Options, class FString& ErrorMessage);
};

// Class TAGame.GameInfo_Entry_TA
// 0x0000 (0x04D8 - 0x04D8)
class AGameInfo_Entry_TA : public AGameInfoBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Entry_TA");
}

return uClassPointer;
};

void PostBeginPlay();
```

```

};

// Class TAGame.LoadingScreen_TA
// 0x00E4 (0x0060 - 0x0144)
class ULoadingScreen_TA : public UObject
{
public:
TArray<class FString> ProtipMessages; // 0x0060 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
TArray<class FString> ProtipMessagesQueue; // 0x0070 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString MaximumPlayerCounts; // 0x0080 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UGFxEngine_TA* GFxEngineRef; // 0x0090 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UOnlineGame_X* OnlineGame; // 0x0098 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FLoadingMovieMapInfo LoadingScreenInfo; // 0x00A0
(0x00A0) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long bUnfairTeams : 1; // 0x0140 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long bComingFromEntry : 1; // 0x0140 (0x0004)
[0x0000000000000000] [0x00000002]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadingScreen_TA");
}

return uClassPointer;
};

void HandlePreLoadMap(class FString MapName);
class FString GetProtipMessageWithIcons(class FString ProtipMessage, class FString
OpenDelimiter, class FString CloseDelimiter, uint8_t InputType);
uint8_t GetProtipInputType();
void OnLargeImageMapLoaded(struct FMapImageLoadResult LoadResult);
void HandleCurrentPlaylistSet(class UOnlineGameAccount_X* InAccount, class
UGameSettingPlaylist_X* NewPlaylist);
void HandlePreLoadReplay(class UReplayManager_TA* ReplayManager, class UReplay_TA*
Replay);
void HandleMaxPlayersChanged(class UOnlineGameJoinGame_X* OnlineGameJoinGame,
int32_t MaxPlayers);
void HandleJoiningGame(class UOnlineGameJoinGame_X* OnlineGameJoinGame);
void HandleOnlineGameInitialized();
void HandlePostLoadMap();
void HandleLoadingUnfairMatch(class APlayerControllerBase_TA* PlayerControllerBase);
class FString GetLocalizedMaximumPlayerCounts(int32_t MaxPlayers);
void HandleMatchSettingsChanged(class AGameEvent_TA* GameEvent, struct

```

```
FCustomMatchSettings NewMatchSettings);
void Init(class UGFxEngine_TA* NewGFxEngine);
};

// Class TAGame.GameInfo_Football_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_Football_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Football_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_FTE_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_FTE_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_FTE_TA");
}

return uClassPointer;
};

void eventPostBeginPlay();
};

// Class TAGame.GameInfo_GameEditor_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_GameEditor_TA : public AGameInfo_Soccar_TA
{
public:

public:
```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_GameEditor_TA");
}

return uClassPointer;
};

void RestartGameEvent();
void PlayerResetTraining();
void SetupPlayer(class APlayerController* NewPlayer);
void eventPostLogin(class APlayerController* NewPlayer);
void eventPostBeginPlay();
};

// Class TAGame.MusicStingersPreviewer_TA
// 0x0034 (0x0060 - 0x0094)
class UMusicStingersPreviewer_TA : public UObject
{
public:
class UProductLoader_TA*           ProductLoader;          // 0x0060 (0x0008)
[0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UAkSoundCue*                 StingerPreviewFadeCue; // 0x0068
(0x0008) [0x0000000000000001] (CPF_Edit)
float                           PlayPreviewDelay;        // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_MusicStingers_TA* AnthemAsset;        // 0x0078
(0x0008) [0x0000008000002000] (CPF_Transient)
struct FProductInstanceId         AnthemId;             // 0x0080 (0x0010)
[0x0000004000002000] (CPF_Transient)
unsigned long                     bHoveringOverSlot : 1; // 0x0090 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long                     bPlaying : 1;        // 0x0090 (0x0004)
[0x0000004000002000] [0x00000002] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicStingersPreviewer_TA");
}

return uClassPointer;
};

bool IsPreviewable();
void ClearPreview();

```

```

void UpdatePreview();
void StopPreview();
void PlayPreview();
void StartPreview();
void HandleProductLoaded(class UProductLoader_TA* InLoader);
void SetPreviewSlotIndex(int32_t PreviewIndex);
void LoadAnthemAsset(int32_t ProductID);
void SetProduct(int32_t ProductID, struct FProductInstanceId OnlineProductId);
void SetPreview(int32_t SlotIndex, struct FProductHashSource ProductSource);
};

// Class TAGame.PostMatchCelebrationPreviewer_TA
// 0x001C (0x0070 - 0x008C)
class UPostMatchCelebrationPreviewer_TA : public UComponent
{
public:
class ACarPreviewActor_TA*           PreviewActor;          // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FName>               PreviewAnims;        // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t                            PreviewAnimIdx;       // 0x0088 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PostMatchCelebrationPreviewer_TA");
}

return uClassPointer;
};

void ResetInvalidAssets(int32_t SlotIndex, int32_t ProductID);
void SetPreviewProduct(int32_t SlotIdx, int32_t ProductID, struct FProductInstanceId OnlinID,
int32_t TeamIdx, unsigned long bUsePrimaryBaseLoadout);
void SetPreviewBaseLoadout();
void ResetPreviewActorMesh();
void ClearPreview();
void PlayNextAnim();
void PlayAnims();
void UpdateMesh(class ACarPreviewActor_TA* _);
bool IsCurrentlyInShop();
void InitPreview();
class USeqEvent_PreviewPostMatchCelebration_TA* GetPMCEvent();
void NotifyPreviewKismet(unsigned long bStart);
void InitVars();
void HandleCarPreviewActorAdded(class UGameShare_TA* GameShare, class
ACarPreviewActor_TA* InPreviewActor);
void Init();
};

```

```

// Class TAGame.GFxData_MenuSequence_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_MenuSequence_TA : public UGFxDataSingleton_X
{
public:
unsigned long           bInMainMenu : 1;           // 0x0098 (0x0004)
[0x0000004040000000] [0x00000001] (CPF_EditInlineNotify)
TArray<struct FMenuSequencePair>      MenuStack;        // 0x00A0
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MenuSequence_TA");
}

return uClassPointer;
};

void OnSequenceChange(struct FName NewSequence);
void HandleSequenceChanged(class UMenuSequence_TA* NewSequence);
void SetMenuSequence(struct FName MenuSequence);
void PopMenuSequence(struct FName MenuID);
void PushMenuSequence(struct FName MenuSequence, struct FName MenuID);
void eventOnShellSet();
bool CheckIsInMainMenu();
bool IsInMainMenu();
};

// Class TAGame.GameInfo_GodBall_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_GodBall_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_GodBall_TA");
}

return uClassPointer;
};

}

```

```
// Class TAGame.GameInfo_Hockey_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_Hockey_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Hockey_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_Items_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_Items_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Items_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_KnockOut_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_KnockOut_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_KnockOut_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_AprilFool_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_AprilFool_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_AprilFool_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_BeachBall_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_BeachBall_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_BeachBall_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_BoomerBall_TA
// 0x0000 (0x0550 - 0x0550)
```

```
class AGameInfo_LTM_BoomerBall_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_BoomerBall_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_Demolition_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_Demolition_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_Demolition_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_DropshotRumble_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_DropshotRumble_TA : public AGameInfo_Breakout_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_DropshotRumble_TA");
}
```

```
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_Eggstra_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_Eggstra_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_Eggstra_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_GForce_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_GForce_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_GForce_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_Moonball_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_Moonball_TA : public AGameInfo_Soccar_TA
{
public:
```

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_Moonball_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.GameInfo_LTM_Pinball_TA  
// 0x0000 (0x0550 - 0x0550)  
class AGameInfo_LTM_Pinball_TA : public AGameInfo_Soccar_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_Pinball_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.GameInfo_LTM_SpeedDemon_TA  
// 0x0000 (0x0550 - 0x0550)  
class AGameInfo_LTM_SpeedDemon_TA : public AGameInfo_Soccar_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_SpeedDemon_TA");  
}  
  
return uClassPointer;
```

```
};

};

// Class TAGame.GameInfo_LTM_SpikeRush_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_SpikeRush_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_SpikeRush_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_LTM_SuperCube_TA
// 0x0000 (0x0550 - 0x0550)
class AGameInfo_LTM_SuperCube_TA : public AGameInfo_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_LTM_SuperCube_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_PerfTest_TA
// 0x0000 (0x04D8 - 0x04D8)
class AGameInfo_PerfTest_TA : public AGameInfoBase_TA
{
public:

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_PerfTest_TA");
}

return uClassPointer;
};

int32_t SortCameraActors(class ACameraActor* A, class ACameraActor* B);
void SelectCamera(int32_t Direction);
void PrevCamera();
void NextCamera();
void GenericPlayerInitialization(class AController* C);
class APlayerController* SpawnPlayerController(struct FVector SpawnLocation, struct FRotator
SpawnRotation);
};

// Class TAGame.PostProcessManager_TA
// 0x0038 (0x00D0 - 0x0108)
class UPostProcessManager_TA : public UPostProcessManager_X
{
public:
uint8_t PostProcessAAType; // 0x00D0 (0x0001)
[0x0000008000002000] (CPF_Transient) DOF_DistanceLimit; // 0x00D4 (0x0004)
float [0x0000000000000000] DOF_InnerRadiusLimit; // 0x00D8 (0x0004)
float [0x0000000000000000] DOF_KernelDefault; // 0x00DC (0x0004)
float [0x0000000000000000] DOF_MaxFarDefault; // 0x00E0 (0x0004)
float [0x0000000000000000] DOF_DistanceExtent; // 0x00E4 (0x0004)
unsigned long bPostProcessedSettingsCached : 1; // 0x00E8
(0x0004) [0x0000000000000000] [0x00000001]
struct FScriptDelegate __EventPostProcessSettingsCached__Delegate; // 0x00F0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PostProcessManager_TA");
}

return uClassPointer;
};

```

```

float GetPostProcessProperty(uint8_t PropertyName);
void SetPostProcessProperty(uint8_t PropertyName, float InValue, unsigned long bSetToDefault);
void SetDOFDistance(float Value);
void NotifyWhenPostProcessSettingsCached(struct FScriptDelegate Callback);
bool IsReplayFXDirtied();
void RestoreReplayFXPostProcessSettings();
void CacheReplayFXPostProcessSettings();
void ApplyPostProcessTypeOverride();
void SetAAType(uint8_t NewType);
void EventPostProcessSettingsCached();
};

// Class TAGame.GameInfo_Season_TA
// 0x0008 (0x0540 - 0x0548)
class AGameInfo_Season_TA : public AGameInfo_TA
{
public:
    class AGameEvent_Soccar_TA*           SoccarArchetype;          // 0x0540
    (0x0008) [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Season_TA");
        }

        return uClassPointer;
    };

    void RestartLevel();
    void RestartPlayer(class AController* aPlayer);
    bool PlayerCanRestart(class APlayerController* aPlayer);
    void eventPostLogin(class APlayerController* NewPlayer);
};

// Class TAGame.GameInfo_TrainingEditor_TA
// 0x0004 (0x0550 - 0x0554)
class AGameInfo_TrainingEditor_TA : public AGameInfo_GameEditor_TA
{
public:
    unsigned long           bCanExportMatinee : 1;          // 0x0550 (0x0004)
    [0x0001000000002000] [0x00000001] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
    }
}

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_TrainingEditor_TA");
}

return uClassPointer;
};

void SetupPlayer(class APlayerController* NewPlayer);
void OnCreateCurrentGame();
};

// Class TAGame.GameInfo_Transition_TA
// 0x0000 (0x04D8 - 0x04D8)
class AGameInfo_Transition_TA : public AGameInfoBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Transition_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameInfo_Tutorial_TA
// 0x0004 (0x0550 - 0x0554)
class AGameInfo_Tutorial_TA : public AGameInfo_Soccar_TA
{
public:
unsigned long          bAutoEntered : 1;           // 0x0550 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameInfo_Tutorial_TA");
}

return uClassPointer;
};

void TutorialHelper();

```

```

void SkipTutorial();
void InitTutorialGameEvent(class AGameEvent_TA* GameEventArchetype, int32_t InDifficulty);
void StartTutorialType(class FString Type, class FString Difficulty);
void SetTutorialDifficulty(int32_t Difficulty);
void eventPostBeginPlay();
};

// Class TAGame.SeqEvent_TutorialHelper_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_TutorialHelper_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_TutorialHelper_TA");
}

return uClassPointer;
};

};

// Class TAGame.GameModesConfig_TA
// 0x0020 (0x0078 - 0x0098)
class UGameModesConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FName> EnabledGameModes; // 0x0078
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<int32_t> ForcedDetailsGroups; // 0x0088 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameModesConfig_TA");
}

return uClassPointer;
};

bool ShouldForceDetailsView(int32_t DetailsGroup);
bool IsGameModeEnabled(struct FName GameMode);
};

```

```

// Class TAGame.GameMusicComponent_TA
// 0x0000 (0x0070 - 0x0070)
class UGameMusicComponent_TA : public UComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GameMusicComponent_TA");
}

return uClassPointer;
};

void Tick(float DeltaTime);
void ShutDown();
void Init();
};

// Class TAGame.TourServerInfo_TA
// 0x0018 (0x0060 - 0x0078)
class UTourServerInfo_TA : public UObject
{
public:
struct FName           InfoName;          // 0x0060 (0x0008)
[0x0001000000000002] (CPF_Const)
int32_t                NumAllowedSpectators; // 0x0068 (0x0004)
[0x0001000000000002] (CPF_Const)
unsigned long           bUsesTieBreakerRules : 1; // 0x006C (0x0004)
[0x0001000000000002] [0x00000001] (CPF_Const)
unsigned long           bRestoreInterruptedMatches : 1; // 0x006C (0x0004)
[0x0001000000000002] [0x00000002] (CPF_Const)
class UClass*           GameUpdateClass;    // 0x0070 (0x0008)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourServerInfo_TA");
}

return uClassPointer;
};

```

```

class UObject* CreateClientReservationMessage(class UReservationBeacon_X* Beacon, struct
FServerReservationData& Reservation, struct FUniqueNetId& PlayerID, struct
FTourServerSettings& TourSettings);
class FString GetTieBreakerRules(struct FTourServerSettings& InSettings);
bool IsTourSet(struct FTourServerSettings& TourSettings);
};

// Class TAGame.PlaylistSettings_TA
// 0x0070 (0x0158 - 0x01C8)
class UPlaylistSettings_TA : public UPlaylistSettings_X
{
public:
TArray<struct FMapSkillSettings> MapSkillRequirements; // 0x0158
(0x0010) [0x0000000000040001] (CPF_Edit | CPF_NeedCtorLink)
class UClass* TourServerInfoClass; // 0x0168 (0x0008)
[0x0001000000000001] (CPF_Edit)
unsigned long bAllowHonorDuelChallenges : 1; // 0x0170 (0x0004)
[0x0001000000000001] [0x00000001] (CPF_Edit)
unsigned long bAllowVoiceChat : 1; // 0x0170 (0x0004)
[0x0001000000000001] [0x00000002] (CPF_Edit)
int32_t DetailsGroup; // 0x0174 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UBackFillPolicy_TA* BackFillPolicy; // 0x0178 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FTeamColor> TeamColors; // 0x0180 (0x0010)
[0x0000000000040000] (CPF_NeedCtorLink)
struct FColorOverride BallSuperSonicColor; // 0x0190 (0x0014)
[0x0000000000000000]
struct FClientLoadoutData LoadoutOverrides[0x2]; // 0x01A8
(0x0020) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlaylistSettings_TA");
}

return uClassPointer;
};

};

// Class TAGame.OverlayInteractionEGS_TA
// 0x0000 (0x00D0 - 0x00D0)
class UOverlayInteractionEGS_TA : public UInteraction
{
public:

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OverlayInteractionEGS_TA");
}

return uClassPointer;
};

bool HandleInputChar(int32_t ControllerId, class FString Character);
bool HandleInputAxis(int32_t ControllerId, struct FName Key, float delta, float DeltaTime,
unsigned long bGamepad);
bool HandleInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float
AmountDepressed, unsigned long bGamepad);
};

// Class TAGame.GarageArchiveSlot_TA
// 0x0008 (0x0060 - 0x0068)
class UGarageArchiveSlot_TA : public UObject
{
public:
class UProductsArchiveSave_TA*           ArchiveSave;          // 0x0060 (0x0008)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GarageArchiveSlot_TA");
}

return uClassPointer;
};

void __GarageArchiveSlot_TA__Construct_0x2(class UClass* _, uint8_t SelectedFilter);
void __GarageArchiveSlot_TA__Construct_0x1(uint8_t SelectedFilter);
void HandleArchivedProductsChanged();
void HandleArchiveSaveLoaded(class UProductsArchiveSave_TA* InArchiveSave);
int32_t GetPreviewQuantity(unsigned long bIgnoreAdditionalFilters);
bool HasArchivedProducts();
bool AttributeSelected();
bool ViewingArchiveSlot();
int32_t ArchiveAttributeArrayIndex();
int32_t ArchiveSlotArrayIndex();
uint8_t GetArchivedProductFilter(uint8_t DefaultArchiveFilter);
void UpdateProducts(uint8_t SelectedFilter, uint8_t DefaultArchiveFilter);
void UpdateSlotVisibility(uint8_t SelectedFilter);
void UpdatePreviewQuantity(uint8_t SelectedFilter);
void UpdateAttributeVisibility(unsigned long bArchiveSlotSelected);

```

```

void OnSlotSelectedChanged(int32_t SlotIndex, unsigned long bSelected, uint8_t SelectedFilter);
void RemoveExclusiveAttribute(uint8_t SelectedFilter);
void eventConstruct();
};

// Class TAGame.ProductAttribute_Archived_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_Archived_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Archived_TA");
}

return uClassPointer;
};

};

// Class TAGame.GarageFavoritedSlot_TA
// 0x0008 (0x0060 - 0x0068)
class UGarageFavoritedSlot_TA : public UObject
{
public:
class UProductsFavoriteSave_TA*           FavoriteSave;          // 0x0060 (0x0008)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GarageFavoritedSlot_TA");
}

return uClassPointer;
};

void __GarageFavoritedSlot_TA__Construct_0x1(uint8_t SelectedFilter);
void HandleFavoritedProductsChanged();
void HandleFavoriteSaveLoaded(class UProductsFavoriteSave_TA* InFavoriteSave);
void UpdateProducts(uint8_t SelectedFilter, uint8_t DefaultFavoriteFilter);
void eventConstruct();
};

```

```
// Class TAGame.GarageSlotDropGroup_TA
// 0x000C (0x00C0 - 0x00CC)
class UGarageSlotDropGroup_TA : public UItemDropGroup_TA
{
public:
    class UTexture*           Icon;          // 0x00C0 (0x0008)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t                  SlotIndex;     // 0x00C8 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GarageSlotDropGroup_TA");
        }

        return uClassPointer;
    }
};

// Class TAGame.GFxData_GarageSlotDropGroup_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_GarageSlotDropGroup_TA : public UGFxData_ItemDropGroup_TA
{
public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_GarageSlotDropGroup_TA");
        }

        return uClassPointer;
    }
};

// Class TAGame.GFxData_GenericNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_GenericNotification_TA : public UGFxData_Notification_TA
{
public:
    static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_GenericNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.GetUploadUrlsMatch_TA
// 0x0048 (0x0060 - 0x00A8)
class UGetUploadUrlsMatch_TA : public UObject
{
public:
class FString MatchGuid; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
uint64_t RecordStartTimestamp; // 0x0070 (0x0008)
[0x0000000000000000]
int32_t Playlist; // 0x0078 (0x0004)
[0x0000000000000000]
TArray<struct FName> Mutators; // 0x0080 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bIsBotMatch : 1; // 0x0090 (0x0004)
[0x0000000000000000] [0x00000001]
TArray<class UGetUploadUrlsPlayer_TA*> Players; // 0x0098
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GetUploadUrlsMatch_TA");
}

return uClassPointer;
};

};

// Class TAGame.GetUploadUrlsPlayer_TA
// 0x0050 (0x0060 - 0x00B0)
class UGetUploadUrlsPlayer_TA : public UObject
{
public:
struct FUniqueNetId PlayerID; // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t ActorID; // 0x00A8 (0x0004)

```

```
[0x0000000000000000] Mu; // 0x00AC (0x0004)
float [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GetUploadUrlsPlayer_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_AdHoc_TA
// 0x0002 (0x0098 - 0x009A)
class UGFxData_AdHoc_TA : public UGFxDataSingleton_X
{
public:
uint8_t AdHocDesiredState; // 0x0098 (0x0001)
[0x0000000400000000] (CPF_EditInlineNotify)
uint8_t AdHocState; // 0x0099 (0x0001)
[0x0000000400000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_AdHoc_TA");
}

return uClassPointer;
};

void HandleDisabledFail(class UError* Error);
void HandleDisabledSuccess();
void DisableAdHoc();
void OnEnabledError();
void OnEnabled();
void HandleEnabledFail(class UError* Error);
void HandleEnabledSuccess();
void EnableAdHoc();
};

// Class TAGame.GFxData_AutoTour_CompletedResults_TA
// 0x00BC (0x0094 - 0x0150)
```

```

class UGFxData_AutoTour_CompletedResults_TA : public UGFxDataRow_X
{
public:
    class UPsyNet_X*           PsyNet;           // 0x0098 (0x0008)
    [0x0001800000000000]
    class UOnlineGameTournaments_TA*   Tournaments;      // 0x00A0
    (0x0008) [0x0001800000000001] (CPF_Edit)
    class UAutoTourConfig_TA*       Config;          // 0x00A8 (0x0008)
    [0x0001800000000000]
    class UAutoTour_TA*          AutoTour;        // 0x00B0 (0x0008)
    [0x0001000000000000]
    uint64_t                     CycleID;         // 0x00B8 (0x0008)
    [0x000100000002000] (CPF_Transient)
    int32_t                      WeekIndex;       // 0x00C0 (0x0004)
    [0x000100000002000] (CPF_Transient)
    unsigned long                 bEliminatedFirstRound : 1; // 0x00C4 (0x0004)
    [0x000100000002000] [0x00000001] (CPF_Transient)
    unsigned long                 bWeeksResultsChanged : 1; // 0x00C4 (0x0004)
    [0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
    unsigned long                 bReceivedResults : 1; // 0x00C4 (0x0004)
    [0x0001000040000000] [0x00000004] (CPF_EditInlineNotify)
    unsigned long                 bFailedToRetrieveResults : 1; // 0x00C4 (0x0004)
    [0x0001000040000000] [0x00000008] (CPF_EditInlineNotify)
    struct FTournamentResult     NewTournamentResult; // 0x00C8
    (0x0010) [0x000100000002000] (CPF_Transient)
    TArray<struct FTournamentResult> OldWeeklyResults; // 0x00D8
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FTournamentResult> NewWeeklyResults; // 0x00E8
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FShopCurrencyInfo> CurrencyRewards; // 0x00F8
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FShopCurrencyInfo> OldWeeklyCurrencies; // 0x0108
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FShopCurrencyInfo> WeeklyCurrencies; // 0x0118
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FShopCurrencyInfo> FinalCurrencies; // 0x0128
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    uint64_t                     TournamentID;      // 0x0138 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t                     ScheduleID;       // 0x0140 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)
    uint64_t                     RejoinTournamentTime; // 0x0148 (0x0008)
    [0x0001000040000000] (CPF>EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_AutoTour_CompletedResults_TA");
        }
    }
}

```

```

return uClassPointer;
};

void __GFxData_AutoTour_CompletedResults_TA__Init_0x3(class URPC_X* RPC);
void __GFxData_AutoTour_CompletedResults_TA__Init_0x2(class
URPC_AutoTour_GetCompletedResult_TA* RPC);
void __GFxData_AutoTour_CompletedResults_TA__Init_0x1(class
UPsyNetService_AutoTourCompletedResultReceived_TA* Service);
void __GFxData_AutoTour_CompletedResults_TA__HandleFailedToRetrieveResults_0x2(class
URPC_X* RPC);
void __GFxData_AutoTour_CompletedResults_TA__HandleFailedToRetrieveResults_0x1(class
URPC_AutoTour_GetCompletedResult_TA* RPC);
int32_t __GFxData_AutoTour_CompletedResults_TA__UpdateTournamentResults_0x2(struct
FTournamentResult A, struct FTournamentResult B);
int32_t __GFxData_AutoTour_CompletedResults_TA__UpdateTournamentResults_0x1(struct
FTournamentResult A, struct FTournamentResult B);
bool __GFxData_AutoTour_CompletedResults_TA__HasEligibleRejoinTournament_0x1(class
UGFxData_AutoTour_TourCard_TA* T);
bool HasEligibleRejoinTournament();
TArray<struct FShopCurrencyInfo> GetFinalCurrencies();
TArray<struct FShopCurrencyInfo> GetWeeklyCurrencies();
TArray<struct FShopCurrencyInfo> GetOldWeeklyCurrencies();
TArray<struct FShopCurrencyInfo> GetCurrencyRewards();
TArray<struct FTournamentResult> GetNewResults();
TArray<struct FTournamentResult> GetOldResults();
struct FTournamentResult GetNewTournamentResult();
bool UpdateTournamentResults(struct FTournamentResult Result, TArray<struct
FTournamentResult>& InOldWeeklyResults);
void SetEndOfTournamentData(struct FTournamentEndData Data);
void OnFailedToRetrieveResults(class UError* InError);
void HandleFailedToRetrieveResults();
void Init(class UTournamentCompletedEvent_TA* InTournamentCompletedEvent);
};

// Class TAGame.RPC_AutoTour_GetCompletedResult_TA
// 0x00C0 (0x00E8 - 0x01A8)
class URPC_AutoTour_GetCompletedResult_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    uint64_t                      TournamentID;      // 0x0130 (0x0008)
    [0x0001004000000000]
    struct FTournamentEndData     EndOfTournamentData; // 0x0138
    (0x0070) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_AutoTour_GetCompletedResult_TA");
        }
    }
};

```

```

}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_AutoTourCompletedResultReceived_TA
// 0x0070 (0x0090 - 0x0100)
class UPsyNetService_AutoTourCompletedResultReceived_TA : public UPsyNetClientService_X
{
public:
struct FTournamentEndData           EndOfTournamentData;          // 0x0090
(0x0070) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.PsyNetService_AutoTourCompletedResultReceived_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_AutoTour_TA
// 0x0090 (0x0098 - 0x0128)
class UGFxData_AutoTour_TA : public UGFxDataSingleton_X
{
public:
class UAutoTourConfig_TA*          Config;                      // 0x0098 (0x0008)
[0x0001000000000000]
class FString                     ScheduleRegionLabel;        // 0x00A0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                           NumWeeksInCycle;           // 0x00B0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                           CycleScore;                 // 0x00B4 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t                           CurrencyID;                // 0x00B8 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
unsigned long                      bDisplayCompletedTourResults : 1; // 0x00BC
(0x0004) [0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
TArray<class UGFxData_AutoTour_TourCard_TA*> TourCards;          // 0x00C0
(0x0010) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UGFxData_AutoTour_Week_TA*> Weeks;                  // 0x00D0
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UGFxData_AutoTour_CompletedResults_TA* EndOfTournamentResults; // 0x00E0 (0x0008)
[0x0001000000002000] (CPF_Transient)

```

```

class FString                                TournamentBanMessage;           // 0x00E8 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                                PartyMembersTournamentBanned;   // 0x00F8
(0x0010) [0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UPsyNet_X*                            PsyNet;                         // 0x0108 (0x0008)
[0x0001800000000000]
class UOnlineGameTournaments_TA*          Tournaments;                   // 0x0110
(0x0008) [0x0001800000000001] (CPF_Edit)
class UAutoTourConfig_TA*                 AutoTourConfig;                // 0x0118 (0x0008)
[0x0001800000000000]
class UAutoTour_TA*                      AutoTour;                      // 0x0120 (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_AutoTour_TA");
}

return uClassPointer;
};

void __GFxData_AutoTour_TA__HandleAutoTourNotFound_0x1();
void __GFxData_AutoTour_TA__HandleAutoTourNotFound_0x2(class UGFxModal_X* Modal);
void EndCompletedTourResultScreen();
static void HandleTournamentCompleted(class UAutoTour_TA* AutoTourGroup, class
UGFxData_AutoTour_TA* GFxAutoTour, class APlayerController_Menu_TA* _, class
UTournamentCompletedEvent_TA* InTourCompletedEvent);
TArray<int32_t> GetMaxTierScores();
void AutoTourResultsUpdated();
void HandleCycleUpdated();
void HandleRegionUpdated();
void HandleSchedulesUpdated();
void HandleAutoTourNotFound(class UPsyNetService_AutoTourNotFound_TA* Service);
void HandleBannedPartyMembers(class
UPsyNetService_AutoTourReceiveBannedPartyMembers_TA* Service);
void HandleReceivedBan(class UPsyNetService_AutoTourReceiveBan_TA* Service);
void HandleConfigChanged();
void hide();
bool IsInformationOutOfDate();
void Show();
void HandleGRISpawned(class AGRI_X* GRI);
void eventOnShellSet();
};

// Class TAGame.GFxData_AutoTour_Week_TA
// 0x001C (0x0094 - 0x00B0)
class UGFxData_AutoTour_Week_TA : public UGFxDataRow_X
{
public:

```

```
int32_t Score; // 0x0098 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
TArray<struct FTournamentResult> Results; // 0x00A0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_AutoTour_Week_TA");
}

return uClassPointer;
};

int32_t __GFxData_AutoTour_Week_TA__Init_0x2(int32_t Sum, struct FTournamentResult Result);
int32_t __GFxData_AutoTour_Week_TA__Init_0x1(struct FTournamentResult A, struct
FTournamentResult B);
TArray<struct FTournamentResult> GetResults();
void Init(TArray<struct FTournamentResult>& InResults);
};

// Class TAGame.PsyNetService_AutoTourReceiveBan_TA
// 0x0008 (0x0090 - 0x0098)
class UPsyNetService_AutoTourReceiveBan_TA : public UPsyNetClientService_X
{
public:
int32_t AutoTournamentBanSeconds; // 0x0090 (0x0004)
[0x0000000000000000]
int32_t MatchmakingBanSeconds; // 0x0094 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_AutoTourReceiveBan_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_AutoTourReceiveBannedPartyMembers_TA
// 0x0010 (0x0090 - 0x00A0)
class UPsyNetService_AutoTourReceiveBannedPartyMembers_TA : public
UPsyNetClientService_X
```

```

{
public:
TArray<struct FUniqueNetId>           BannedTeamMembers;          // 0x0090
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.PsyNetService_AutoTourReceiveBannedPartyMembers_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_AutoTourNotFound_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_AutoTourNotFound_TA : public UPsyNetClientService_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_AutoTourNotFound_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Matchmaking_TA
// 0x00E4 (0x0114 - 0x01F8)
class UGFxData_Matchmaking_TA : public UGFxData_OnlineMatch_TA
{
public:
class FString           NoPlaylistSelectedMessage;      // 0x0118 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           NoRegionSelectedMessage;       // 0x0128 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           PartyTooBigMessage;           // 0x0138 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           LeavePenaltyMessage;         // 0x0148 (0x0010)

```

```

[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               MatchmakingBanMessage;           // 0x0158 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               PartySkillTooDistant;          // 0x0168 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               PartyMemberNotOriginalAppOwner; // 0x0178
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                               PartyMemberNotLoggedIn;         // 0x0188 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
TArray<struct FGfxRegion>                 Regions;                      // 0x0198 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t                                     TotalPopulation;              // 0x01A8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
uint8_t                                      MatchmakingViewTab;           // 0x01AC (0x0001)
[0x0000000040000000] (CPF>EditInlineNotify)
unsigned long                                bRankedPlaylistsEnabled : 1;   // 0x01B0 (0x0004)
[0x0000004040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long                                bMultiSelectEnabled : 1;      // 0x01B0 (0x0004)
[0x0000004040000000] [0x00000002] (CPF>EditInlineNotify)
int32_t                                      SeasonEndDays;               // 0x01B4 (0x0004)
[0x0000004040000000] (CPF>EditInlineNotify)
int32_t                                      SeasonEndHours;              // 0x01B8 (0x0004)
[0x0000004040000000] (CPF>EditInlineNotify)
int32_t                                      SeasonEndMinutes;            // 0x01BC (0x0004)
[0x0000004040000000] (CPF>EditInlineNotify)
class UMatchmakingSettingsSave_TA*          MatchmakingSave;              // 0x01C0
(0x0008) [0x0000004000000000]
class UPlayMenuSave_TA*                     PlayMenuSave;                // 0x01C8 (0x0008)
[0x0000004000000000]
TArray<struct FMessageUpdate>              MessageUpdates;              // 0x01D0
(0x0010) [0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)
int32_t                                      CurrentMessageUpdateIndex;    // 0x01E0 (0x0004)
[0x0000000000002000] (CPF_Transient)
class URankedConfig_X*                     RankedConfig;                // 0x01E8 (0x0008)
[0x0000800000000001] (CPF>Edit)
class UMatchmakingViewFilters_TA*           TabFilters;                  // 0x01F0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Matchmaking_TA");
}

return uClassPointer;
};

void __GFxData_Matchmaking_TA__OnShellSet_0x1(unsigned long _);
void UpdateSelectedRegions();
void UpdateSeasonTimeRemaining();

```

```

void HandleDisableCrossPlayChange(class UOnlineGameParty_X* PartyObject);
uint8_t GetStatusOwnerValue();
void CancelSearch();
void CancelNextDisplayMessage();
void DisplayNextMessage();
void AddMessageUpdate(int32_t DelayFromStart, class FString MessageKey);
void PopulateMessageUpdates();
TArray<struct FName> GetAccessiblePlaylists();
bool StartMatchmaking(int32_t ForcedPlaylistID);
bool IsRankedView(uint8_t Tab);
void SetMatchmakingViewTab(uint8_t InMatchmakingViewTab);
void SetRegionSelection(int32_t Row, unsigned long bSelected);
void DeselectAllRegions();
void HandleRegionsSynced(class UGFxData_Regions_TA* RegionsData);
void SetPlaylistSelectionByPlaylistID(int32_t PlaylistId, unsigned long bSelected);
void SetPlaylistSelection(int32_t Row, unsigned long bSelected);
void DeselectAllPlaylists(unsigned long bRankedOnly);
void HandleRankedConfigChanged();
void InitPlaylists();
void HandleGetPlaylistPopulations(class UOnlineGamePopulation_X* Population);
void UpdatePopulation();
void ToggleMultiSelectMode();
void SetViewingMatchmaking(unsigned long bViewingMatchmaking);
int32_t GetMaxTeamSize();
bool GetNotLoggedInPartyRestrictionError(class FString& OutError);
bool GetCompetitivePartyRestrictionError(class FString& OutError);
class FString GetOnlineStatusError();
class FString ReplMatchmakingBanTime(class FString InBanMessage);
bool IsBannedFromMatchmaking();
float GetEstimatedQueueTimeSeconds();
void HandlePlayMenuSaveLoaded(class UPlayMenuSave_TA* InPlayMenuSave);
void HandleMatchmakingSaveLoaded(class UMatchmakingSettingsSave_TA*
InMatchmakingSave);
void eventOnShellSet();
};

// Class TAGame.GFxModal_CurrencyMessage_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_CurrencyMessage_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_CurrencyMessage_TA");
}

return uClassPointer;
};

```

```

};

// Class TAGame.TourCheckInWindowWatcher_TA
// 0x0038 (0x0060 - 0x0098)
class UTourCheckInWindowWatcher_TA : public UObject
{
public:
    class UTourConfig_TA*           Config;           // 0x0060 (0x0008)
    [0x0001800000000001] (CPF_Edit)
    class UTourSettings_TA*         NextTournament;   // 0x0068 (0x0008)
    [0x0001004000002000] (CPF_Transient)
    uint64_t                         CheckInWindowSeconds; // 0x0070 (0x0008)
    [0x0001000000002000] (CPF_Transient)
    uint64_t                         ActivityWindowSeconds; // 0x0078 (0x0008)
    [0x0001000000002000] (CPF_Transient)
    struct FScriptDelegate          __EventWindowChange__Delegate; // 0x0080
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.TourCheckInWindowWatcher_TA");
        }

        return uClassPointer;
    }

    bool CanShowActivityNotifications(uint64_t StartTime, uint64_t CurrentTime);
    bool CanCheckIn(uint64_t StartTime, uint64_t CurrentTime);
    float GetTimeToNextWindow();
    void HandleCheckInWindowEvent();
    void ClearCheckInTimer();
    void UpdateCheckInTimer(uint64_t CurrentTime, uint64_t NextEventTime);
    uint64_t GetWindowOpenTime(uint64_t StartTime);
    static uint64_t StaticGetWindowOpenTime(uint64_t StartTime, uint64_t WindowSeconds);
    static uint64_t GetNextWindowEventTime(uint64_t CurrentTime, uint64_t StartTime, uint64_t WindowSeconds);
    static class UTourSettings_TA* FindNextWindow(uint64_t CurrentTime, uint64_t WindowSeconds, TArray<class UTourSettings_TA*>& Tournaments);
    void HandleConfigChange();
    void StopWatch();
    void StartWatch(TArray<class UTourSettings_TA*> Tournaments, uint64_t CurrentTime);
    void EventWindowChange(class UTourCheckInWindowWatcher_TA* Watcher, class UTourSettings_TA* Tournament);
};

// Class TAGame.GFxData_BanManager_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_BanManager_TA : public UGFxDataSingleton_X

```

```
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_BanManager_TA");  
}  
  
return uClassPointer;  
};  
  
void OnBanMessagesSynced();  
void HandleSyncedBanStatus(class UBanSync_TA* BanSync);  
void RemoveBanMessages();  
void eventOnRemoved();  
void eventOnShellSet();  
};  
  
// Class TAGame.GFxData_BanMessage_TA  
// 0x0004 (0x0094 - 0x0098)  
class UGFxData_BanMessage_TA : public UGFxDataRow_X  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_BanMessage_TA");  
}  
  
return uClassPointer;  
};  
  
class UBanMessage_X* GetBanMessage();  
bool IsPermanentlyBanned();  
bool IsBanned();  
TArray<class FString> GetCitations();  
int32_t GetSecondsRemaining();  
void HandleBanEnded(class UBanMessage_X* BanMessage);  
void SetMessage(class UBanMessage_X* BanMessage);  
};  
  
// Class TAGame.GFxData_BlogTile_TA  
// 0x0018 (0x0094 - 0x00AC)  
class UGFxData_BlogTile_TA : public UGFxDataRow_X
```

```

{
public:
class UTexture*           ImageTexture;           // 0x0098 (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t                   TextureSizeX;          // 0x00A0 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t                   TextureSizeY;          // 0x00A4 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
unsigned long              bNewInfo : 1;          // 0x00A8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_BlogTile_TA");
}

return uClassPointer;
};

void __GFxDATA_BlogTile_TA_SetBlogTile_0x1(class UTexture2DDynamic* BlogTexture);
void SetIsNewInfo(unsigned long bInNewInfo);
class UGFxDATA_BlogTile_TA* SetBlogTile(class UBlogTile_X* InBlogTile);
};

// Class TAGame.GFxData_BlueprintGarage_TA
// 0x0028 (0x0098 - 0x00C0)
class UGFxDATA_BlueprintGarage_TA : public UGFxDATASingleton_X
{
public:
class UOnlineProductStoreSet_TA*      BlueprintGarageSet;      // 0x0098
(0x0008) [0x000100004082008] (CPF>ExportObject | CPF>Transient | CPF>Component |
CPF>EditInline)
TArray<struct FBlueprintPreviewProduct> BlueprintPreviewProducts; // 0x00A0
(0x0010) [0x000100000402000] (CPF>Transient | CPF>NeedCtorLink)
class UOnlineProduct_TA*             PendingBlueprintProduct; // 0x00B0
(0x0008) [0x0001000000000000]
class USaveData_TA*                 SaveData;                // 0x00B8 (0x0008)
[0x000100000002000] (CPF>Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_BlueprintGarage_TA");
}
}

```

```

return uClassPointer;
};

void __GFxData_BlueprintGarage_TA__OnShellSet_0x2(class FString _);
void __GFxData_BlueprintGarage_TA__OnShellSet_0x1(class USaveGameManager_TA* M, class
USaveData_TA* S, class UError* E);
bool __GFxData_BlueprintGarage_TA__CreatePreviewProduct_0x1(struct FOnlineProductAttribute
Attribute);
bool __GFxData_BlueprintGarage_TA__OwnsRequiredProduct_0x1(class UProduct_TA* P);
void OnForceOpenFTUE();
void OnBuildFail();
void OnRevealFail();
void OnBuildSuccess(struct FProductHashID Product);
void OnRevealSuccess(struct FProductHashID RevealedBlueprint);
void HandleBuildFail(class URPC_X* RPC);
void HandleRevealFail(class URPC_X* RPC);
void HandleBuildSuccess(class URPC_BuildBlueprint_TA* RPC);
void HandleRevealSuccess(class URPC_RevealBlueprint_TA* RPC);
struct FProductHashID CreateProductFromBlueprint(struct FOnlineProductData ProductToAdd,
struct FProductInstanceId BlueprintToRemove);
void ReturnPendingBlueprint();
void StorePendingBlueprint(class UOnlineProduct_TA* Blueprint);
bool OwnsRequiredProduct(struct FProductHashID BlueprintHashID);
bool IsAlreadyOwned(struct FProductHashID BlueprintHashID);
void Build(struct FProductHashID BlueprintHashID);
void Reveal(struct FProductHashID BlueprintHashID);
void Preview(struct FProductHashID BlueprintHashID);
bool PreviewProductSeriesIDChanged(struct FBlueprintPreviewProduct PreviewProduct);
TArray<struct FProductHashID> GetSimilarHashIDs(class UOnlineProduct_TA* OnlineProduct);
void CreatePreviewProduct(struct FProductHashID BlueprintHashID, int32_t Index);
class UOnlineProduct_TA* GetOrCreatePreviewProduct(struct FProductHashID BlueprintHashID);
void eventOnRemoved();
void eventOnShellSet();
};


```

```

// Class TAGame.GFxData_MainMenu_TA
// 0x0050 (0x0098 - 0x00E8)
class UGFxData_MainMenu_TA : public UGFxDataSingleton_X
{
public:
float PreviewCameraRotationScale; // 0x0098 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FName RequestedDLCName; // 0x009C (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t RequestedSubIdx; // 0x00A4 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FName QueuedName; // 0x00A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t QueuedSubIdx; // 0x00B0 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FName UISoundState_AudioPreview; // 0x00B4 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName UISoundState_MusicPreview; // 0x00BC (0x0008)
[0x0000000000000002] (CPF_Const)

```

```

struct FName           UISoundState_CurrentPreview;          // 0x00C4 (0x0008)
[0x0000000000000000]
struct FScriptDelegate      ProductLoadedCallback;          // 0x00D0 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MainMenu_TA");
}

return uClassPointer;
};

void ClearPostMatchPreview();
void InitPostMatchPreview();
void OnSettingsAutoUpdated(uint8_t Reason);
static void HandleSkillDecayed(class UGFxShell_TA* GFxShell, class ULocalPlayer_TA*
LocalPlayer, class USkillDecayedEvent_TA* SkillDecayedEvent);
static void MainMenuAdded(class UGFxData_MainMenu_TA* GFxMainMenu);
void OnEnteredMainMenu();
void HandleTopMenuChange(struct FName PrevMenu, struct FName TopMenu, unsigned long
bClearingStack);
class UShowroomMetrics_TA* GetShowroomMetrics();
void SetLockPreviewActor(unsigned long bLock);
void PopCurrentSoundState();
void PushCurrentSoundState(struct FName SoundState);
bool SetPreviewProduct(int32_t SlotIndex, struct FProductHashID HashID, unsigned long
bUseBaseLoadout);
void SetPreviewBaseLoadout();
void OnRotatePreviewFinished();
void RotatePreviewCamera(float X, float Y);
void OnPremiumCarFinished();
void SetPremiumCarFromBundle(struct FName PackID, int32_t SubIdx);
void SetPremiumDLCPack(struct FName PackID, int32_t SubIdx);
class UProfile_TA* GetProfile();
void eventOnShellSet();
};

// Class TAGame.RPC_RevealBlueprint_TA
// 0x0098 (0x00E8 - 0x0180)
class URPC_RevealBlueprint_TA : public URPC_X
{
public:
struct FUniqueNetId        PlayerID;                      // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FProductInstanceId   InstanceID;                   // 0x0130 (0x0010)
[0x0000000000000000]
struct FOnlineProductData    ProductData;                 // 0x0140 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RevealBlueprint_TA");
}

return uClassPointer;
};

class URPC_RevealBlueprint_TA* SetInstanceId(struct FProductInstanceId InInstanceId);
class URPC_RevealBlueprint_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_BuildBlueprint_TA
// 0x00B0 (0x00E8 - 0x0198)
class URPC_BuildBlueprint_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;                      // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FProductInstanceId      InstanceID;                   // 0x0130 (0x0010)
[0x0000000000000000]
int32_t                       ExpectedPrice;                // 0x0140 (0x0004)
[0x0000000000000000]
struct FOnlineProductData     ProductData;                 // 0x0148 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCurrency>     WalletCurrencies;          // 0x0188 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_BuildBlueprint_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_CarColors_TA
// 0x0060 (0x0098 - 0x00F8)
class UGFxData_CarColors_TA : public UGFxDataSingleton_X
{
public:

```

```

TArray<struct FGFxTeamColor>           Custom;          // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor>           Team0;          // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor>           Team1;          // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor>           Clubs;          // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor>           PlayerBanners; // 0x00D8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor>           PlayerVanity; // 0x00E8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_CarColors_TA");
}

return uClassPointer;
};

void Init();
void eventOnShellSet();
};

// Class TAGame.GFxData_CarKnockOut_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_CarKnockOut_TA : public UGFxDataSingleton_X
{
public:
int32_t           LivesRemaining;          // 0x0098 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t           DodgesRemaining;         // 0x009C (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
int32_t           JumpsRemaining;          // 0x00A0 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
float             StunlockAlpha;          // 0x00A4 (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
uint8_t            StunlockType;           // 0x00A8 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
float             GrabbingAlpha;          // 0x00AC (0x0004)
[0x0001000040000000] (CPF>EditInlineNotify)
class ACar_KnockOut_TA* ActiveGrabbedCar; // 0x00B0 (0x0008)
[0x000100000002000] (CPF>Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_CarKnockOut_TA");
}

return uClassPointer;
};

static void SpectatorHUDDestroyed(class UGFxData_CarKnockOut_TA* CarKnockOut, class AGFxHUD_Spectator_TA* SpectatorHUD);
void SpectatorViewTargetChanged(class APRI_TA* PreviousTarget, class APRI_TA* CurrentTarget, class AGFxHUD_Spectator_TA* SpectatorHUD);
static void SpectatorHUDSpawned(class UGFxData_CarKnockOut_TA* CarKnockOut, class AGFxHUD_Spectator_TA* SpectatorHUD);
void HandleGrabbingComplete();
void HandleGrabbingTimeChanged(class AStunlock_TA* Stunlock);
void HandleGrabbedCarChanged(class ACar_KnockOut_TA* Car, class ACar_KnockOut_TA* GrabbedCar);
void HandleCarRemoved(class ACar_KnockOut_TA* Car);
void HandleCarAdded(class ACar_KnockOut_TA* Car);
void HandleStunlockComplete();
void HandleStunlockTimeChanged(class AStunlock_TA* Stunlock);
void HandleStunlockAdded(class AStunlock_TA* Stunlock);
void HandleLivesChanged(class APRI_KnockOut_TA* PRI);
void HandlePRISet(class APRI_KnockOut_TA* PRI);
void HandleDoubleJumpComponentRemoved(class ACarComponent_DoubleJump_KO_TA* DoubleJumpComponent);
void HandleDoubleJumpAirCountChanged(class ACarComponent_DoubleJump_KO_TA* DoubleJumpComponent);
void HandleDoubleJumpComponentSet(class ACarComponent_DoubleJump_KO_TA* DoubleJumpComponent);
void HandleDodgeComponentRemoved(class ACarComponent_Dodge_KO_TA* DodgeComponent);
void HandleDodgeAirCountChanged(class ACarComponent_Dodge_KO_TA* DodgeComponent);
void HandleDodgeComponentSet(class ACarComponent_Dodge_KO_TA* DodgeComponent);
void eventOnShellSet();
};

// Class TAGame.GFxHUD_Spectator_TA
// 0x0060 (0x0540 - 0x05A0)
class AGFxHUD_Spectator_TA : public AGFxHUD_TA
{
public:
class UGFxData_ReplayViewer_TA* ViewerData; // 0x0540
(0x0008) [0x0000000000002000] (CPF_Transient)
float ZoomSpeed; // 0x0548 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ZoomClickTime; // 0x054C (0x0004)
[0x0000000000000001] (CPF_Edit)
class AGFxHUD_TA* GameHUD; // 0x0550 (0x0008)
[0x0000000000002000] (CPF_Transient)
class APRI_TA* TargetPlayerPRI; // 0x0558 (0x0008)
[0x0000000000002000] (CPF_Transient)

```

```
unsigned long          bZoomIn : 1;           // 0x0560 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long          bZoomOut : 1;          // 0x0560 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
float                 ZoomInTime;           // 0x0564 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                 ZoomOutTime;          // 0x0568 (0x0004)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate __EventTargetPlayerPRIChanged__Delegate; // 0x0570
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __OpenMenuWithKeycodeDelegate__Delegate; // 0x0588 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_Spectator_TA");
}

return uClassPointer;
};

void __GFxHUD_Spectator_TA__InitGFx_0x1(class AGameEvent_Soccar_TA* InGameEvent);
static class FString __GFxHUD_Spectator_TA__GetFocusPlayerNames_0x3(class UGFxData_PRI_TA* Data);
static int32_t __GFxHUD_Spectator_TA__GetFocusPlayerNames_0x2(class UGFxData_PRI_TA* L,
class UGFxData_PRI_TA* R);
static bool __GFxHUD_Spectator_TA__GetFocusPlayerNames_0x1(class UGFxData_PRI_TA*
Data);
void CycleNameplate();
void HandleCanVoteForfeitChanged(class AGameEvent_TA* InGameEvent);
class ACar_TA* GetFocusCar();
class APRI_TA* GetPRIFromPlayerIDString(class FString PlayerIDString);
void HandleViewedPlayerTeamChanged(class APRI_X* InPRI);
void ViewCycle(int32_t Direction);
void ViewPrev();
void ViewNext();
void ViewFly();
void ViewAutoCam();
void ViewDefault();
void SetFocusActorString(class FString FocusActorString);
void OnSpectatorHUDSpawned();
void HandleEliminated(int32_t SpectateDelay);
void TargetPlayerPRIChanged(class FString NewTargetPlayerIDString);
void ViewPlayer(int32_t Team, int32_t Player);
void OpenReplayFXMenu();
void OpenCameraMenu();
void CycleCamera();
void OpenFocusMenu();
void CycleFocus();
```

```

static TArray<class FString> GetFocusPlayerNames(TArray<class UGFxData_PRI_TA*>& PRIList);
void OpenMenuWithKeycode(struct FName Action, struct FScriptDelegate Callback);
void OpenMenuWithKeycodeDelegate(int32_t KeyCode, int32_t MouseButton);
void UpdateHudVisibility();
void CycleHUD();
bool ProcessZoomPress(unsigned long bPressed, int32_t DeltaFOV, float& Time);
void ZoomOut(unsigned long bPressed);
void ZoomIn(unsigned long bPressed);
void AdjustZoom(float DeltaFOV);
void TickZoom(float DeltaTime, unsigned long bZoom, float Speed, float& Time);
void eventTick(float DeltaTime);
void SetGameEvent(class AGameEvent_TA* InGameEvent);
class UGFxData_PRI_TA* CreateGFxPRIData();
void HandleReservationsUpdated(class AGRI_X* GRI);
void eventDestroyed();
void InitGFx();
void EventTargetPlayerPRIChanged(class APRI_TA* PreviousTarget, class APRI_TA*
CurrentTarget, class AGFxHUD_Spectator_TA* SpectatorHUD);
};

// Class TAGame.GFxData_ChallengeObject_TA
// 0x0004 (0x0094 - 0x0098)
class UGFxData_ChallengeObject_TA : public UGFxDataRow_X
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeObject_TA");
}

return uClassPointer;
};

void eventOnRemoved();
void HandleChallengeObjectRemoved(class UListenObject_TA* Challenge);
void SetChallengeProxy(class UListenObject_TA* Challenge);
};

// Class TAGame.GFxData_ChallengeBase_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_ChallengeBase_TA : public UGFxData_ChallengeObject_TA
{
public:
class UGFxData_ChallengeReward_TA*           GFxReward;           // 0x0098
(0x0008) [0x0001000000000000]
TArray<class UGFxData_ChallengeCurrency_TA*>   GFxCurrency;        // 0x00A0
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<class UGFxData_ChallengeRequirement_TA*> GFxRequirement;    //

```

```

0x00B0 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
class UGFxModal_X*                               ModalProcessing;           // 0x00C0 (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeBase_TA");
}

return uClassPointer;
};

void __GFxData_ChallengeBase_TA__OnRemoved_0x2(class
UGFxData_ChallengeRequirement_TA* R);
void __GFxData_ChallengeBase_TA__OnRemoved_0x1(class UGFxData_ChallengeCurrency_TA*
C);
class UChallenge_TA* GetChallengeProxy();
bool IsAvailable();
void TimeElapsed();
void HandleRewardCollected(class UChallengeManager_TA* Manager, class UChallenge_TA*
Challenge, TArray<class UOnlineProduct_TA*> OnlineProductRewards, struct FOnlineReward&
Reward);
void HandleRewardsRequestComplete(class URPC_Challenge_RequestReward_TA* RPC);
void CollectRewards();
void ClearNewInfolcon();
void CloseProcessingModal();
void eventOnRemoved();
void SetChallengeProxy(class UListenObject_TA* Challenge);
};

// Class TAGame.GFxData_ChallengeCurrency_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeCurrency_TA : public UGFxData_ChallengeObject_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeCurrency_TA");
}

return uClassPointer;
};

```

```

};

// Class TAGame.GFxData_ChallengeRequirement_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeRequirement_TA : public UGFxData_ChallengeObject_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeRequirement_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ChallengeReward_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeReward_TA : public UGFxData_ChallengeObject_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeReward_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_Challenge_RequestReward_TA
// 0x004C (0x00E8 - 0x0134)
class URPC_Challenge_RequestReward_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t                      Id;                 // 0x0130 (0x0004)
[0x0000000000000000]

```

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_RequestReward_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_Challenge_RequestReward_TA* SetChallengeID(int32_t ChallengeID);  
class URPC_Challenge_RequestReward_TA* SetPlayerID(struct FUniqueNetId InPlayerId);  
};  
  
// Class TAGame.GFxData_Challenge_TA  
// 0x0000 (0x00C8 - 0x00C8)  
class UGFxData_Challenge_TA : public UGFxData_ChallengeBase_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Challenge_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.GFxData_ChallengeBucket_TA  
// 0x0000 (0x00C8 - 0x00C8)  
class UGFxData_ChallengeBucket_TA : public UGFxData_ChallengeBase_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeBucket_TA");  
}
```

```

return uClassPointer;
};

};

// Class TAGame.GFxData_ChallengeConfig_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ChallengeConfig_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeConfig_TA");
}

return uClassPointer;
};

void HandleEventConfigChanged(class UChallengeConfig_TA* Config);
void eventOnShellSet();
};

// Class TAGame.GFxData_ChallengeFolder_TA
// 0x005C (0x0094 - 0x00F0)
class UGFxData_ChallengeFolder_TA : public UGFxDataRow_X
{
public:
unsigned long          bSelected : 1;           // 0x0098 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bNotifyNewInfo : 1;        // 0x0098 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long          bNormalRewardsAvailable : 1; // 0x0098 (0x0004)
[0x0001000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long          bPremiumRewardsAvailable : 1; // 0x0098 (0x0004)
[0x0001000040000000] [0x00000008] (CPF_EditInlineNotify)
class UGFxData_ChallengeFolder_TA*      GFxParentFolder; // 0x00A0
(0x0008) [0x0001000000000000]
TArray<class UGFxData_ChallengeFolder_TA*>    GFxSubFolders; // 0x00A8
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)
class UClass*            SubFolderClass; // 0x00B8 (0x0008)
[0x0001000000000000]
struct FScriptDelegate      __EventSelected__Delegate; // 0x00C0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate      __EventRefreshed__Delegate; // 0x00D8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeFolder_TA");
}

return uClassPointer;
};

void __GFxData_ChallengeFolder_TA__RemoveSubFolders_0x1(class
UGFxData_ChallengeFolder_TA* Folder);
void __GFxData_ChallengeFolder_TA__CreateSubFolders_0x1(class UChallengeFolder_TA*
Folder);
bool __GFxData_ChallengeFolder_TA__RefreshNotifies_0x3(class UChallenge_TA* C);
bool __GFxData_ChallengeFolder_TA__RefreshNotifies_0x2(class UChallenge_TA* C);
bool __GFxData_ChallengeFolder_TA__RefreshNotifies_0x1(class UChallenge_TA* C);
class UChallengeManager_TA* GetChallengeManager();
void RefreshNotifies(TArray<class UChallenge_TA*> Challenges);
TArray<class UChallenge_TA*> GetSortedChallenges(class UClass* Filter);
class UChallengeFolder_TA* GetFolderProxy();
void CreateSubFolders();
void RemoveSubFolders();
class UGFxData_ChallengeFolder_TA* AddSubFolder(class UClass* FolderClass, class
UChallengeFolder_TA* ChallengeFolder);
void HandleSubFolderRefreshed(class UGFxData_ChallengeFolder_TA* GFxFolder);
void HandleSubFolderSelected(class UGFxData_ChallengeFolder_TA* GFxFolder);
void Unselect();
void Select();
void SetFolderProxy(class UChallengeFolder_TA* ChallengeFolder);
void eventOnRemoved();
void EventRefreshed(class UGFxData_ChallengeFolder_TA* GFxFolder);
void EventSelected(class UGFxData_ChallengeFolder_TA* GFxFolder);
};

// Class TAGame.GFxData_ChallengeManager_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_ChallengeManager_TA : public UGFxDataSingleton_X
{
public:
class FString ActiveFolderCodeName; // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint8_t ActiveChallengeFilter; // 0x00A8 (0x0001)
[0x0001000040000000] (CPF_EditInlineNotify)
class UChallengeManager_TA* ChallengeManager; // 0x00B0
(0x0008) [0x00010000000000]
TArray<class UGFxData_ChallengePage_TA*> GFxChallengePages; // 
0x00B8 (0x0010) [0x0001000000400000] (CPF_NeedCtorLink)
TArray<class UClass*> ChallengeFilters; // 0x00C8 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeManager_TA");
}

return uClassPointer;
};

bool IsRewardAvailable(int32_t ChallengeID);
bool TriggerCompletedChallengeNotification(int32_t ChallengeID);
void HandlePageSelected(class UGFxData_ChallengeFolder_TA* GFxFolder);
void RemovePages();
class UGFxData_ChallengePage_TA* CreatePage(class UChallengeFolder_TA* Page, class
UClass* ChallengeFilter);
void SetActiveFolder(class FString FolderCodeName, uint8_t ChallengeFilter);
void HandleChallengesSynced(class UChallengeDefaultManager_TA*
ChallengeDefaultManager);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_ChallengePage_TA
// 0x0008 (0x00F0 - 0x00F8)
class UGFxData_ChallengePage_TA : public UGFxData_ChallengeFolder_TA
{
public:
class UClass*           ChallengeFilter;          // 0x00F0 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengePage_TA");
}

return uClassPointer;
};

void __GFxData_ChallengePage_TA__CreateSubFolders_0x1();
TArray<class UChallenge_TA*> GetSortedChallenges(class UClass* Filter);
void CreateSubFolders();
void RemoveGFxChallengeData();
class UGFxData_ChallengeFolder_TA* AddSubFolder(class UClass* FolderClass, class
UChallengeFolder_TA* ChallengeFolder);
void Select();
void eventOnRemoved();
};

```

```

// Class TAGame.GFxData_ChallengeTab_TA
// 0x001C (0x00F0 - 0x010C)
class UGFxData_ChallengeTab_TA : public UGFxData_ChallengeFolder_TA
{
public:
    class UChallengeManager_TA*           ChallengeManager;          // 0x00F0
    (0x0008) [0x0001000000000000]
    int32_t                               TotalNormalChallenges;   // 0x00F8 (0x0004)
    [0x0001000040000000] (CPF_EditInlineNotify)
    int32_t                               CompletedNormalChallenges; // 0x00FC (0x0004)
    [0x0001000040000000] (CPF_EditInlineNotify)
    int32_t                               TotalPremiumChallenges;   // 0x0100 (0x0004)
    [0x0001000040000000] (CPF_EditInlineNotify)
    int32_t                               CompletedPremiumChallenges; // 0x0104 (0x0004)
    [0x0001000040000000] (CPF_EditInlineNotify)
    unsigned long                         bHasVisibleChallenges : 1; // 0x0108 (0x0004)
    [0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_ChallengeTab_TA");
        }

        return uClassPointer;
    }

void __GFxData_ChallengeTab_TA__OnShellSet_0x4(class UChallengeManager_TA* Manager);
void __GFxData_ChallengeTab_TA__OnShellSet_0x3(class UChallengeManager_TA* Manager,
class UChallenge_TA* Challenge, TArray<class UOnlineProduct_TA*> OnlineProductRewards,
struct FOnlineReward& Reward);
void __GFxData_ChallengeTab_TA__OnShellSet_0x2(class UChallengeManager_TA* Manager,
class UChallenge_TA* Challenge);
void __GFxData_ChallengeTab_TA__OnShellSet_0x1(class UChallengeManager_TA* Manager,
class UChallenge_TA* Challenge);
void __GFxData_ChallengeTab_TA__RefreshChallenges_0x1(class UChallenge_TA* C);
bool __GFxData_ChallengeTab_TA__RefreshNotifies_0x1(class UChallenge_TA* C);
bool HasBucketChallenge();
void RefreshNotifies(TArray<class UChallenge_TA*> Challenges);
TArray<class UChallenge_TA*> GetSortedChallenges(class UClass* Filter);
void RemoveGFxChallengeData();
void AddGFxChallenge(class UChallenge_TA* Challenge);
void HandleChallengeAdded(class UChallengeManager_TA* Manager, class UChallenge_TA* Challenge);
void RefreshChallenges();
void Select();
void eventOnRemoved();
void eventOnShellSet();
};

```

```

// Class TAGame.ProfileQuickChatSave_TA
// 0x0010 (0x00C8 - 0x00D8)
class UProfileQuickChatSave_TA : public USaveObject_TA
{
public:
TArray<struct FName>           QuickChatBindings;          // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfileQuickChatSave_TA");
}

return uClassPointer;
};

void __ProfileQuickChatSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void ResetQuickChatBindings();
void ChangeQuickChatBinding(int32_t Index, struct FName Message);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
};

// Class TAGame.SeqAct_CinematicIntroStartNextSeq_TA
// 0x0020 (0x0160 - 0x0180)
class USeqAct_CinematicIntroStartNextSeq_TA : public USequenceAction
{
public:
int32_t           CurrentSequenceIndex;        // 0x0160 (0x0004)
[0x0000000000000000]
struct FScriptDelegate      __EventActivated__Delegate;    // 0x0168
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_CinematicIntroStartNextSeq_TA");
}

return uClassPointer;
};

void eventActivated();
void EventActivated(class USeqAct_CinematicIntroStartNextSeq_TA* SeqAct);
};

```

```
// Class TAGame.GFxData_ClientPerformanceStats_TA
// 0x0004 (0x0098 - 0x009C)
class UGFxData_ClientPerformanceStats_TA : public UGFxDataSingleton_X
{
public:
unsigned long           bIsGameRecordingEnabled : 1;          // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClientPerformanceStats_TA");
}

return uClassPointer;
};

void eventOnShellSet();
};

// Class TAGame.PsyNetService_ClubsInvite_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_ClubsInvite_TA : public UPsyNetClientService_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_ClubsInvite_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxModal_Error_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Error_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Error_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ClubsConfig_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ClubsConfig_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClubsConfig_TA");
}

return uClassPointer;
};

void HandleClubConfigChanged(class UClubsConfig_TA* Config);
void eventOnShellSet();
};

// Class TAGame.GFxData_ClubServerBrowser_TA
// 0x001C (0x0098 - 0x00B4)
class UGFxData_ClubServerBrowser_TA : public UGFxDataSingleton_X
{
public:
class UOnlineClubServerList_X*           ServerList;          // 0x0098 (0x0008)
[0x0001000000000000]
TArray<struct FGFxClubServer>           Servers;            // 0x00A0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long                           bRefreshing : 1;    // 0x00B0 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ClubServerBrowser_TA");
}

return uClassPointer;
};

void JoinServer(int32_t RowNumber);
void OnRefreshed(class FString Error);
struct FGFxClubServer CreateGFxClubServer(class UClubServerResult_X* Server);
void UpdateGFxServers(class UError* Error);
void Refresh();
};

// Class TAGame.GFxData_ServerBrowser_TA
// 0x0058 (0x0098 - 0x00F0)
class UGFxData_ServerBrowser_TA : public UGFxDataSingleton_X
{
public:
class ULocalPlayer_TA* TAPlayer; // 0x0098 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString ServerName; // 0x00A0 (0x0010)
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString Password; // 0x00B0 (0x0010)
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FServerResult> Results; // 0x00C0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UGFxModal_X* ProcessingModal; // 0x00D0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString JoiningServerName; // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UServerBrowserSettingsSave_TA* SaveData; // 0x00E8
(0x0008) [0x00000000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ServerBrowser_TA");
}

return uClassPointer;
};

void HandleCancelSearch(class UGFxModal_X* Modal);
void HandleCancelJoin(class UGFxModal_X* Modal);
void HandleJoinGameComplete(unsigned long bSuccess, class FString FailReason);
void JoinServer(class FString InServerName, struct FServerReservationData Reservation, struct FJoinMatchSettings JoinSettings);
void HandleReservation(struct FServerReservationData& Reservation);
void HandleSearchComplete(TArray<struct FServerResult>& InResults);

```

```

void HandleSearchError(class FString msg);
void CancelJoin();
void CancelSearch();
void StartSearch();
void SetPassword(class FString InPassword);
void SetServerName(class FString InServerName);
void JoinServerByNamePassword(class FString CustomServerName, class FString
CustomServerPassword, unsigned long bClubServer);
void HandleSaveDataLoad(class UServerBrowserSettingsSave_TA* InSaveData);
void eventOnShellSet();
class UOnlineGame_X* GetOnlineGame();
};

// Class TAGame.GFxData_ConnectionStats_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_ConnectionStats_TA : public UGFxDataSingleton_X
{
public:
class UConnectionQualityConfig_TA*           ConnectionQualityConfig;          // 0x0098
(0x0008) [0x0000800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ConnectionStats_TA");
}

return uClassPointer;
};

void AcceptedHelpModal(class UGFxModal_X* Modal);
void CheckConnectionQualityHelp();
void HandleTrackerAttached(class APlayerController_TA* PC, class
UClientConnectionTracker_TA* Tracker);
void HandleControllerReceived(class UPlayer* InPlayer);
void HandleGRISpawned(class AGRI_X* GRI);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.RPC_MicroTransactions_GetContainerDropTable_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_MicroTransactions_GetContainerDropTable_TA : public URPC_X
{
public:
TArray<struct FRPCCContainerDropTable>      ContainerDrops;                // 0x00E8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_MicroTransactions_GetContainerDropTable_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProfilePCSave_TA
// 0x002C (0x00C8 - 0x00F4)
class UProfilePCSave_TA : public USaveObject_TA
{
public:
TArray<struct FPlayerBinding> PCBindings; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
float MouseSensitivity; // 0x00D8 (0x0004)
[0x0000000000000000]
float MouseXDeadZone; // 0x00DC (0x0004)
float MouseYDeadZone; // 0x00E0 (0x0004)
float MouseXDeadZoneAir; // 0x00E4 (0x0004)
float MouseYDeadZoneAir; // 0x00E8 (0x0004)
float KeyboardAxisBlendTime; // 0x00EC (0x0004)
unsigned long bAirPitchSafetyEnabled : 1; // 0x00F0 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfilePCSave_TA");
}

return uClassPointer;
};

void __ProfilePCSave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void OnCreate();
};

```

```

// Class TAGame.GFxData_Credits_TA
// 0x0060 (0x0098 - 0x00F8)
class UGFxData_Credits_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FLocalizedDepartmentData> LocalizedDepartmentGroups; // 0x0098 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FLocalizedPostCreditData> LocalizedCopyrightGroups; // 0x00A8 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FLocalizedPostCreditData> LocalizedMusicGroups; // 0x00B8 (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FDepartmentData> DepartmentGroups; // 0x00C8 (0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FPostCreditData> CopyrightGroups; // 0x00D8 (0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FPostCreditData> MusicGroups; // 0x00E8 (0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Credits_TA");
}

return uClassPointer;
};

void __GFxData_Credits_TA__LoadCredits_0x1(struct FLocalizedPostCreditData LocalizedPostCredit);
void AddToPostCreditData(struct FLocalizedPostCreditData LocalizedPostCredit, TArray<struct FPostCreditData>& Out_PostCredits);
void AddToDepartmentData(class FString LocalizedDepartmentTitle, struct FLocalizedNamaData LocalizedNameInfo);
void UnloadCredits();
void MarkCreditsViewed();
void LoadCredits();
};

// Class TAGame.GFxData_CrumbTrails_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_CrumbTrails_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FCrumbTrailName> TrailNames; // 0x0098 (0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UCrumbTrails_TA* CrumbsMgr; // 0x00A8 (0x0008) [0x0001800000000000]

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_CrumbTrails_TA");
}

return uClassPointer;
};

struct FCrumbTrailName __GFxData_CrumbTrails_TA__UpdateCrumbsNames_0x1(struct FName N);
bool CanActivateTrail(struct FName TrailId);
void ActivateTrail(struct FName TrailId);
void UpdateCrumbsNames();
void HandleCrumbChange(class UCrumbTrails_TA* CrumTrails);
void HandleSaveAdded(class UCrumbTrailSave_TA* InSave);
void eventOnShellSet();
};

// Class TAGame.GFxData_CustomTeams_TA
// 0x0060 (0x0098 - 0x00F8)
class UGFxData_CustomTeams_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFxCustomTeamSettings> Settings; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint8_t SettingsType; // 0x00A8 (0x0001)
[0x0000000040000000] (CPF>EditInlineNotify)
TArray<struct FGFxTeamColor> TeamColors; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxTeamColor> CustomColors; // 0x00C0
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class ULocalPlayer_TA* TAPlayer; // 0x00D0 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t PreviewTeam; // 0x00D8 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UCarPreviewSet_TA* PreviewSet; // 0x00E0 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FClubColorSet PreviewColors[0x2]; // 0x00E8 (0x0010)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_CustomTeams_TA");
}

return uClassPointer;
}

```

```

};

bool InMainMenu();
void UpdateCarColors(class ACarPreviewActor_TA* PreviewActor, unsigned long bClubColors);
void HandleSetProfile(class ACarPreviewActor_TA* Car);
void RestoreStadiumColors();
void InitStadiumColors();
void ResetPreviewColors();
void SetCustomColorID(int32_t Team, int32_t ColorID);
void PreviewCustomColorID(int32_t Team, int32_t ColorID);
void SetTeamColorID(int32_t Team, int32_t ColorID);
void PreviewTeamColorID(int32_t Team, int32_t ColorID);
void ClearTeamName(int32_t Team);
void SetTeamName(int32_t Team, class FString TeamName);
void SetSettingsType(uint8_t Type);
void SaveColors(int32_t Team);
void InitPreviewColors();
void InitColorTables();
struct FGFxCustomTeamSettings CreateTeamSettings(int32_t TeamIdx);
void InitTeamSettings();
void eventOnShellSet();
};

// Class TAGame.GFxData_DateTime_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxDATA_DateTime_TA : public UGFxDATA_Singleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_DateTime_TA");
}

return uClassPointer;
};

struct FDateTimeStruct AddSeconds(struct FDateTimeStruct InDateTime, int32_t Amount);
struct FDateTimeStruct AddMinutes(struct FDateTimeStruct InDateTime, int32_t Amount);
struct FDateTimeStruct AddHours(struct FDateTimeStruct InDateTime, int32_t Amount);
struct FDateTimeStruct AddDays(struct FDateTimeStruct InDateTime, int32_t Amount);
struct FDateTimeStruct AddMonths(struct FDateTimeStruct InDateTime, int32_t Amount);
struct FDateTimeStruct AddYears(struct FDateTimeStruct InDateTime, int32_t Amount);
struct FDateTimeStruct UtcToLocalTime(uint64_t UtcSeconds);
int32_t UtcSecondsUntil(uint64_t UtcSeconds);
struct FDateTimeStruct UtcNow();
struct FDateTimeStruct Now();
};

```

```
// Class TAGame.ShowroomMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UShowroomMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShowroomMetrics_TA");
}

return uClassPointer;
};

void PurchaseItemInShowRoom(struct FName ItemName, int32_t ItemIndex);
void ViewItemInStore(struct FName ItemName, int32_t ItemIndex);
void ViewItemInShowroom(struct FName ItemName, int32_t ItemIndex, int32_t SubItemIndex);
void ShowroomExit();
void ShowroomEnter();
};

// Class TAGame.GFxData_DynamicLogos_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_DynamicLogos_TA : public UGFxDataSingleton_X
{
public:
class UDynamicLogosConfig_TA*           DynamicLogosConfig;          // 0x0098
(0x0008) [0x0001800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_DynamicLogos_TA");
}

return uClassPointer;
};

void HandleLogoImageChanged(class UTexture* InLogoImage);
bool IsDynamicLogoLoaded();
class UTexture* GetLogoImage();
bool IsDynamicLogosEnabled();
void HandleInterfaceChanged();
void eventOnShellSet();
};
```

```

// Class TAGame.GFxData_EngagementEventsConfig_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_EngagementEventsConfig_TA : public UGFxDataSingleton_X
{
public:
    class UEngagementEventsConfig_TA*           EngagementEvents;          // 0x0098
    (0x0008) [0x0000800000000000]
    class UNotificationSave_TA*                 NotificationSave;         // 0x00A0 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    TArray<struct FName>                      ValidMenus;              // 0x00A8 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    unsigned long                            bEndNotificationShownThisSession : 1; // 0x00B8
    (0x0004) [0x0000000000002000] [0x00000001] (CPF_Transient)
    class UEngagementEventNotification_TA*     CurrentNotification;      // 0x00C0
    (0x0008) [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_EngagementEventsConfig_TA");
        }

        return uClassPointer;
    };

    bool __GFxData_EngagementEventsConfig_TA__HandleNotification_0x1(struct FNotificationStatus N);
    bool GetStartNotificationShown(uint8_t Type);
    void SetStartNotificationShown(uint8_t Type);
    void GetEventTimeRemaining();
    void HandleTopMenuChanged(struct FName PrevMenu, struct FName TopMenu, unsigned long bClearingStack);
    bool GetLocalization(TArray<struct FNotificationStatus> InNotifications, int32_t NotificationType,
    class FString& OutTitle, class FString& OutBody, class FString& OutAnim);
    void HandleEngagementEventNotificationHidden(class UEngagementEventNotification_TA* Notification);
    void HandleNotification();
    void HandleNotificationSaveUpdated();
    void HandleNotificationSaveAdded(class UNotificationSave_TA* InNotificationSave);
    void eventOnShellSet();
    void HandleConfigChanged();
};

// Class TAGame.GFxData_EOSGameClipsConfig_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_EOSGameClipsConfig_TA : public UGFxDataSingleton_X
{
public:
    class UEOSGameClipsConfig_TA*             GameClipsConfig;          // 0x0098

```

(0x0008) [0x0001800000000000]

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSGameClipsConfig_TA");  
}  
  
return uClassPointer;  
};  
  
void HandleGameClipsConfigChanged();  
};  
  
// Class TAGame.GFxData_EOSGameClipsController_TA  
// 0x0060 (0x0098 - 0x00F8)  
class UGFxData_EOSGameClipsController_TA : public UGFxDataSingleton_X  
{  
public:  
class UEOSGameClipsManager_TA*           GameClipsManager;          // 0x0098  
[0x0008] [0x00180400000000]  
class UEOSGameClipsController_TA*         GameClipsController;       // 0x00A0  
[0x0008] [0x00100400000000]  
class FString                         StartClippingMessage;      // 0x00A8 (0x0010)  
[0x0001004000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)  
class FString                         ClipUploadedMessage;       // 0x00B8 (0x0010)  
[0x0001004000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)  
class FString                         ClipFailedMessage;        // 0x00C8 (0x0010)  
[0x0001004000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)  
class FString                         ClipsUploadingMessage;    // 0x00D8 (0x0010)  
[0x0001004000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)  
class FString                         ClipAfterReplayMessage;   // 0x00E8 (0x0010)  
[0x0001004000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSGameClipsController_TA");  
}  
  
return uClassPointer;  
};  
  
void __GFxData_EOSGameClipsController_TA__InitGameClipsController_0x1(class UPlayer* P);  
void __GFxData_EOSGameClipsController_TA__HandleEOSGameClipsControllerChanged_0x1();  
bool ShouldShowGoalReplayClippingButton();
```

```

void RemoveMaskedArea(uint64_t MaskedAreaId);
uint64_t AddMaskedArea(float InTopLeftX, float InTopLeftY, float InBottomRightX, float
InBottomRightY);
static bool IsMaskedAreaValid(float InTopLeftX, float InTopLeftY, float InBottomRightX, float
InBottomRightY);
static class FString GetMaskAsString(float InTopLeftX, float InTopLeftY, float InBottomRightX,
float InBottomRightY);
void AddHUDMessage(class FString InChatMessage);
void HandleErrorOccurred(class UErrorType* InErrorType);
void HandleBindingsChanged(class UPlayerInput_X* PlayerInput);
void HandleClipStatusChanged(int32_t InClipId, uint8_t InNewClipStatus);
void HandleEOSGameClipsControllerChanged();
void InitGameClipsController();
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_EOSGameClipsLocalPlayer_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_EOSGameClipsLocalPlayer_TA : public UGFxDataSingleton_X
{
public:
class UEOSGameClipsManager_TA*           GameClipsManager;          // 0x0098
(0x0008) [0x0001804000000000]
class UEOSGameClipsConfig_TA*            GameClipsConfig;          // 0x00A0
(0x0008) [0x0001804000000000]
class AGameEvent_TA*                    CurrentGameEvent;         // 0x00A8 (0x0008)
[0x0001804000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSGameClipsLocalPlayer_TA");
}

return uClassPointer;
};

void __GFxData_EOSGameClipsLocalPlayer_TA__GameClipsManagerChanged_0x1();
void SetLinkAccountPopupShown(unsigned long bLinkAccountPopupShown);
bool ShouldShowLinkAccountPopup();
bool ShouldAllowExternalLinks();
bool ShouldShowInSettingsMenu();
bool IsAccountLinked();
bool IsUserEligibleForClipping();
bool IsAccountInCubinedMode();
class FString GetEpicAccountId();
void GameEventChanged();
void GameClipsManagerChanged();
void eventOnRemoved();

```

```

void eventOnShellSet();
};

// Class TAGame.GFxData_EOSGameClipsMaskArea_TA
// 0x0024 (0x0098 - 0x00BC)
class UGFxData_EOSGameClipsMaskArea_TA : public UGFxDataSingleton_X
{
public:
    class UEOSGameClipsManager_TA*           GameClipsManager;          // 0x0098
    (0x0008) [0x0001804000000000]
    TArray<struct FGameClipsMaskArea>        ActiveMaskAreas;          // 0x00A0
    (0x0010) [0x000100000400000] (CPF_NeedCtorLink)
    class UCanvas*                          DrawableCanvas;            // 0x00B0 (0x0008)
    [0x0001000000000000]
    struct FColor                           DebugDrawColor;           // 0x00B8 (0x0004)
    [0x0001000000000002] (CPF_Const)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_EOSGameClipsMaskArea_TA");
        }
    }

    return uClassPointer;
};

void __GFxData_EOSGameClipsMaskArea_TA__DrawDebugMaskAreas_0x1(struct
FGameClipsMaskArea MaskArea);
bool IsDebugEnabled();
void OnMaskedAreaRemoved(uint64_t MaskedAreaId, float InTopLeftX, float InTopLeftY, float
InBottomRightX, float InBottomRightY);
void OnMaskedAreaAdded(uint64_t MaskedAreaId, float InTopLeftX, float InTopLeftY, float
InBottomRightX, float InBottomRightY);
void RemoveMaskedArea(uint64_t MaskedAreaId);
uint64_t AddMaskedArea(float InTopLeftX, float InTopLeftY, float InBottomRightX, float
InBottomRightY);
static bool IsMaskedAreaValid(float InTopLeftX, float InTopLeftY, float InBottomRightX, float
InBottomRightY);
static class FString GetMaskAsString(float InTopLeftX, float InTopLeftY, float InBottomRightX,
float InBottomRightY);
void DrawMaskArea(struct FGameClipsMaskArea InMaskArea);
void DrawDebugMaskAreas(class AHUD* InHUD);
void HandleMaskStatusChanged(uint64_t InMaskAreaHandle, struct FGameClipsMaskArea
InMaskArea, uint8_t InNewMaskStatus);
void HandleGameClipsManagerChanged();
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_ESportConfig_TA

```

```
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_ESportConfig_TA : public UGFxDataSingleton_X
{
public:
class UESportConfig_TA*           ESportEvents;          // 0x0098 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ESportConfig_TA");
}

return uClassPointer;
};

void DownloadImage(class FString URL);
void HandleConfigChanged();
};

// Class TAGame.GFxData_Eula_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_Eula_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Eula_TA");
}

return uClassPointer;
};

bool RequiresAcceptance();
void Accepted();
};

// Class TAGame.GFxData_EventsPageBrowser_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_EventsPageBrowser_TA : public UGFxDataSingleton_X
{
public:
unsigned long          bShowButton : 1;          // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
```

```
class UUIConfig_TA* UIConfig; // 0x00A0 (0x0008)
[0x0000800000000001] (CPF_Edit)
class UEventsPageConfig_TA* EventsConfig; // 0x00A8 (0x0008)
[0x0000800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_EventsPageBrowser_TA");
}

return uClassPointer;
};

void Close();
void Open(int32_t BrowserWidth, int32_t BrowserHeight);
void HandleEventsConfigChanged();
};

// Class TAGame.PsyNetService_FacelSearchStarted_TA
// 0x0004 (0x0090 - 0x0094)
class UPsyNetService_FacelSearchStarted_TA : public UPsyNetClientService_X
{
public:
int32_t PartnerID; // 0x0090 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_FacelSearchStarted_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Features_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_Features_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FName> DeprecatedPermanentFeatures; // 0x0098
(0x0010) [0x00000000400001] (CPF_Edit | CPF_NeedCtorLink)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Features_TA");
}

return uClassPointer;
};

void eventOnShellSet();
};

// Class TAGame.GFxData_FirstTimeExperience_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_FirstTimeExperience_TA : public UGFxDataSingleton_X
{
public:
struct FName CurrentCheckpoint; // 0x0098 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FName CurrentGroup; // 0x00A0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long bActive : 1; // 0x00A8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
class UFirstTimeExperienceManager_TA* FTEManager; // 0x00B0
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_FirstTimeExperience_TA");
}

return uClassPointer;
};

void HandleFTEChange(class UFirstTimeExperienceManager_TA* InFTEManager, unsigned long
bIsActive, struct FName GroupName, struct FName CheckpointName);
void LoadFTEMatch(class FString MatchType);
void StreamLoadLevel(class UClass* InSeqClass);
void LoadFTELevel(class FString InType, unsigned long bAutoEntered);
void LoadFTEMatch_Part2();
void LoadFTEMatch_Part1();
bool HasOtherFTEBlockers(struct FName GroupName);
bool TryStartFTEGroup(struct FName GroupName, unsigned long bIgnoreCompletion);
void CheckpointTentativelyCompleted(struct FName InCheckpoint);
void CheckpointCompleted(struct FName InCheckpoint, unsigned long bHasError);

```

```

void HandleManagerCreated(class UFirstTimeExperienceManager_TA* InFTEManager);
void eventOnShellSet();
};

// Class TAGame.TravelManager_TA
// 0x0060 (0x0060 - 0x00C0)
class UTravelManager_TA : public UObject
{
public:
class FString MapName; // 0x0060 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bWorkshopMap : 1; // 0x0070 (0x0004)
[0x0000000000000000] [0x00000001]
class FString GameInfoClass; // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class FString> GameTags; // 0x0088 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class FString> Options; // 0x0098 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __GetMapData__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TravelManager_TA");
}

return uClassPointer;
};

void Travel(class FString InCommand, class APlayerController* InCommandUser);
void Open(class APlayerController* InCommandUser);
void Start(class APlayerController* InCommandUser);
class FString BuildURL(struct FScriptDelegate GetMapDataCallback);
class UMapData_TA* GetMapData(struct FName RequestedMapName);
class UTravelManager_TA* AddServerCommand(class FString ServerCommand);
class UTravelManager_TA* ClearOptions();
class UTravelManager_TA* AddOptions(class FString InCommaSeparatedOptionsList);
class UTravelManager_TA* AddOptionsArray(TArray<class FString> InOptions);
class UTravelManager_TA* ClearGameTags();
class UTravelManager_TA* AddGameTags(class FString InCommaSeparatedTagsList);
class UTravelManager_TA* AddGameTagsArray(TArray<class FString> InGameTags);
TArray<class FString> TrimStrings(TArray<class FString> InArray);
class UTravelManager_TA* SetGameInfoClass(class FString InGameInfoClass);
class UTravelManager_TA* SetWorkshopMap(unsigned long bInWorkshopMap);
class UTravelManager_TA* SetMapName(class FString inMapName);
static class UTravelManager_TA* TravelManager();
};

```

```
// Class TAGame.GFxData_Freeplay_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_Freeplay_TA : public UGFxDataSingleton_X
{
public:
class UFreeplayCommands_TA*           FreeplayCommands;          // 0x0098
(0x0008) [0x0001800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Freeplay_TA");
}

return uClassPointer;
};

bool CanUseFreeplayCommands();
};

// Class TAGame.GFxData_FreeToPlayRewards_TA
// 0x001C (0x0098 - 0x00B4)
class UGFxData_FreeToPlayRewards_TA : public UGFxDataSingleton_X
{
public:
uint8_t             LegacyStatus;          // 0x0098 (0x0001)
[0x000000040000000] (CPF_EditInlineNotify)
TArray<struct FCompensationItem> CompensationItems;          // 0x00A0
(0x0010) [0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long        bIsLegacySet : 1;           // 0x00B0 (0x0004)
[0x000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_FreeToPlayRewards_TA");
}

return uClassPointer;
};

void __GFxData_FreeToPlayRewards_TA__HandleSaveDataLoaded_0x1(class
UOnlineStorageSyncManager_TA* _);
struct FCompensationItem ConvertToProductDrop(struct FOnlineProductData InProductData);
void NotifyStatusChange();
void HandleSyncSuccess();
```

```

void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*
MySaveData, class UError* Error);
void eventOnShellSet();
};

// Class TAGame.PersonaSave_TA
// 0x0050 (0x00C8 - 0x0118)
class UPersonaSave_TA : public USaveObject_TA
{
public:
TArray<struct FPersonaDataId> RecentPlayers; // 0x00C8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPlayerGameID> RecentGameIDs; // 0x00D8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FPlayerLoadoutData> ObservedPlayerLoadouts; // 0x00E8
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
uint8_t PresenceState; // 0x00F8 (0x0001)
[0x0008000000000000]
struct FScriptDelegate __PresenceState__ChangeNotify; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PersonaSave_TA");
}

return uClassPointer;
};

void __PresenceState__ChangeNotifyFunc();
void SetObservedLoadouts(TArray<struct FPlayerLoadoutData>& VanityLoadouts);
void SetRecentPlayers(TArray<struct FPersonaDataId>& Recents, TArray<struct FPlayerGameID>& Games);
};

// Class TAGame.OtherPlayerProfile_TA
// 0x0020 (0x0070 - 0x0090)
class UOtherPlayerProfile_TA : public UComponent
{
public:
class UPlayerAvatarBorder_TA* PlayerBorder; // 0x0070 (0x0008)
[0x00000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UGFxData_PlayerAvatarBorder_TA* GFxPlayerBorder; // 0x0078
(0x0008) [0x0000000000000000]
class UPlayerBanner_TA* PlayerBanner; // 0x0080 (0x0008)
[0x00000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UGFxData_PlayerBanner_TA* GFxPlayerBanner; // 0x0088
(0x0008) [0x0000000000000000]

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OtherPlayerProfile_TA");
}

return uClassPointer;
};

void Clear(class UGFxShell_X* Shell);
void LoadPlayerBanner(struct FOnlineProductData BannerProductData, struct FUniqueNetId
PlayerID, class UGFxShell_X* Shell);
void LoadPlayerBorder(struct FOnlineProductData BorderProductData, struct FUniqueNetId
PlayerID, class UGFxShell_X* Shell);
};

// Class TAGame.GFxData_PlayerBanner_TA
// 0x000C (0x0094 - 0x00A0)
class UGFxData_PlayerBanner_TA : public UGFxDataRow_X
{
public:
class UPlayerBanner_TA*           PlayerBanner;          // 0x0098 (0x0008)
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerBanner_TA");
}

return uClassPointer;
};

void EquipColor(int32_t InColorID);
void EquipProduct(struct FProductHashID HashID);
class FString GetPlayerName();
struct FUniqueNetId GetPlayerID();
void SetBanner(class UPlayerBanner_TA* InBanner);
static class UGFxData_PlayerBanner_TA* Create(class UGFxShell_X* InShell, class
UPlayerBanner_TA* InBanner);
static class UGFxData_PlayerBanner_TA* CreateWithPlayerName(class UGFxShell_X* InShell,
class UPlayerBanner_TA* InBanner, class FString InPlayerName);
static class UGFxData_PlayerBanner_TA* CreateWithPlayerID(class UGFxShell_X* InShell, class
UPlayerBanner_TA* InBanner, struct FUniqueNetId InPlayerId);
};
```

```

// Class TAGame.GFxData_PlayerAvatarBorder_TA
// 0x000C (0x0094 - 0x00A0)
class UGFxData_PlayerAvatarBorder_TA : public UGFxDataRow_X
{
public:
class UPlayerAvatarBorder_TA*           PlayerBorder;          // 0x0098 (0x0008)
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerAvatarBorder_TA");
}

return uClassPointer;
};

void EquipColor(int32_t InColorID);
void EquipProduct(struct FProductHashID HashID);
struct FUniqueNetId GetPlayerID();
void SetBorder(class UPlayerAvatarBorder_TA* InBorder);
static class UGFxData_PlayerAvatarBorder_TA* Create(class UGFxShell_X* InShell, class
UPlayerAvatarBorder_TA* InBorder, struct FUniqueNetId InPlayerId);
};

// Class TAGame.GFxData_FriendsHiddenManager_TA
// 0x0018 (0x0060 - 0x0078)
class UGFxData_FriendsHiddenManager_TA : public UObject
{
public:
class UGFxData_PersonaGroup_TA*           HiddenPresenceGroup; // 0x0060
(0x0008) [0x0000000000000000]
TArray<class UPersona_TA*>               HiddenFriends;      // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_FriendsHiddenManager_TA");
}

return uClassPointer;
};

void RefreshHiddenGroup();
void HandlePersonaHiddenChanged(class UPersona_TA* Persona);

```

```

void HandlePersonasInitialized(class UPersonas_TA* _);
void InitHiddenFriends();
};

// Class TAGame.GFxData_PersonaGroup_TA
// 0x0019 (0x0094 - 0x00AD)
class UGFxData_PersonaGroup_TA : public UGFxDataRow_X
{
public:
class FString           GroupTitle;          // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                 GroupPresenceld;    // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t                 GroupOrigin;        // 0x00AC (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PersonaGroup_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_ReportPlayer_TA
// 0x0020 (0x00E8 - 0x0108)
class URPC_ReportPlayer_TA : public URPC_X
{
public:
TArray<struct FReportedPlayerInfo>      Reports;          // 0x00E8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
class FString           GameID;            // 0x00F8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ReportPlayer_TA");
}

return uClassPointer;
};

```

```

class URPC_ReportPlayer_TA* SetGameID(class FString InGameID);
class URPC_ReportPlayer_TA* SetReports(TArray<struct FReportedPlayerInfo> InReports);
};

// Class TAGame.GFxData_GameEvent_TA
// 0x00A4 (0x0098 - 0x013C)
class UGFxData_GameEvent_TA : public UGFxDataSingleton_X
{
public:
int32_t TimeRemaining; // 0x0098 (0x0004)
[0x000000040000000] (CPF_EditInlineNotify)
int32_t TimeToNextMatch; // 0x009C (0x0004)
[0x000000040000000] (CPF_EditInlineNotify)
int32_t TimeWaitingForPlayersRemaining; // 0x00A0 (0x0004)
[0x000000040000000] (CPF_EditInlineNotify)
unsigned long bOverTime : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bOnline : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bUnfairTeams : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long bForfeit : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bNoContest : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000010] (CPF_EditInlineNotify)
unsigned long bServerTraveling : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000020] (CPF_EditInlineNotify)
unsigned long bRankedMatch : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000040] (CPF_EditInlineNotify)
unsigned long bSolo : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000080] (CPF_EditInlineNotify)
unsigned long bIsBotMatch : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000100] (CPF_EditInlineNotify)
unsigned long bForfeitVoteAllowed : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000200] (CPF_EditInlineNotify)
unsigned long bHasLeaveMatchPenalty : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000400] (CPF_EditInlineNotify)
unsigned long bCanVoteToForfeit : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00000800] (CPF_EditInlineNotify)
unsigned long bViewingEndGameReplay : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00001000] (CPF_EditInlineNotify)
unsigned long bShowIntroScene : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00002000] (CPF_EditInlineNotify)
unsigned long bDisableTeamSelectScreen : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00004000] (CPF_EditInlineNotify)
unsigned long bStayAsPartyButtonVisible : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00008000] (CPF_EditInlineNotify)
unsigned long bStayAsPartyButtonEnabled : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00010000] (CPF_EditInlineNotify)
unsigned long bServerShutdown : 1; // 0x00A4 (0x0004)
[0x000000040000000] [0x00020000] (CPF_EditInlineNotify)
class FString MatchState; // 0x00A8 (0x0010)
[0x000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FName GameState; // 0x00B8 (0x0008)

```

```

[0x0000000040000000] (CPF_EditInlineNotify)
struct FName MatchType; // 0x00C0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FName EventType; // 0x00C8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ReplayScoredBy; // 0x00D0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ReplayAssistedBy; // 0x00D4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float ReplaySpeed; // 0x00D8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float ReplayTime; // 0x00DC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t WinningTeam; // 0x00E0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t TieBreakDecision; // 0x00E4 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t MaxPlayers; // 0x00E8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t MaxScore; // 0x00EC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString PlaylistName; // 0x00F0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FName PlaylistId; // 0x0100 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString ServerName; // 0x0108 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString PartyPlaylistName; // 0x0118 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString MutatorsString; // 0x0128 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t SeriesGames; // 0x0138 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_GameEvent_TA");
}

return uClassPointer;
};

bool ShouldShowPodiumUI();
void SetIsViewingEndgameReplay(unsigned long Value);
void OnSkipScoreIntro();
void PostMatchRequeue();
void OnCloseScoreboard();
void OnOpenScoreboard();
void OnCloseMidgameMenu();

```

```

void OnOpenMidgameMenu();
void OnTickerEvent(class FString EventName, class UTexture* Texture, int32_t ReceiverIndex,
int32_t VictimIndex);
void OnGameMessage(struct FName Type, class FString Message);
void OnStatEvent(class FString EventName, int32_t Points, class UTexture* Texture, struct
FName Sound);
void StayAsPartyVoteBegin();
void ForceStartPlay();
void SetReadyToPlay(unsigned long bReady, unsigned long bReadyAllLocalPlayers);
};

// Class TAGame.GFxData_ProductSlot_TA
// 0x0054 (0x0094 - 0x00E8)
class UGFxData_ProductSlot_TA : public UGFxDataRow_X
{
public:
unsigned long          bEnabled : 1;           // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bCanUnequip : 1;        // 0x0098 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
struct FName            SlotName;             // 0x009C (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString           Label;                // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString           PluralLabel;          // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UTexture*         Icon;                // 0x00C8 (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t                 SlotIndex;            // 0x00D0 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
struct FProductHashID   EquippedHashID0;      // 0x00D4 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
struct FProductHashID   EquippedHashID1;      // 0x00D8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
struct FProductHashID   EquippedPack;         // 0x00DC (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
int32_t                 DefaultProductID;     // 0x00E0 (0x0004)
[0x0000000000000000]
int32_t                 MatchesNeededToUnlock; // 0x00E4 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductSlot_TA");
}

return uClassPointer;
};

```

```
void SetupProductSlot(class UProductSlot_TA* Slot);
};

// Class TAGame.GFxData_GarageProductSlot_TA
// 0x0000 (0x00E8 - 0x00E8)
class UGFxData_GarageProductSlot_TA : public UGFxData_ProductSlot_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_GarageProductSlot_TA");
}

return uClassPointer;
};

void SetupGarageProductSlot(class UProductSlot_TA* Slot, class ULoadoutSet_TA* LoadoutSet,
int32_t PreviewTeam);
};

// Class TAGame.ICustomizableSlot
// 0x0000 (0x0060 - 0x0060)
class UICustomizableSlot : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ICustomizableSlot");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_IntroCarSelect_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_IntroCarSelect_TA : public UGFxDataSingleton_X
{
public:
class UGFxModal_X* ModalProcessing; // 0x0098 (0x0008)
[0x00100000002000] (CPF_Transient)
```

```
class FString           CarBodyName;          // 0x00A0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UGFxData_FirstTimeExperience_TA*      FTE;           // 0x00B0
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_IntroCarSelect_TA");
}

return uClassPointer;
};

void CarsInPosition();
void ToggleCarSelectMode(unsigned long bEnable);
void SelectCar();
void HandleFocusChanged(class UCameraState_IntroMenu_TA* CameraState);
void ShiftFocusLeft();
void ShiftFocusRight();
class UCameraState_IntroMenu_TA* GetCameraState();
void SetCameraState(struct FName StateName);
void HandleIntroCarsInPosition(class AGameInfo_GFxMenu_TA* GameInfoMenu);
void eventOnShellSet();
};

// Class TAGame.IntroCarTemplates_TA
// 0x0010 (0x0060 - 0x0070)
class UIintroCarTemplates_TA : public UObject
{
public:
TArray<struct FEditorIntroCar>           IntroCars;          // 0x0060 (0x0010)
[0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IntroCarTemplates_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_CarSelectStart_TA
```

```
// 0x0004 (0x017C - 0x0180)
class USeqEvent_CarSelectStart_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_CarSelectStart_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ItemShopNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ItemShopNotification_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ItemShopNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ItemShopNotificationsConfig_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_ItemShopNotificationsConfig_TA : public UGFxDataSingleton_X
{
public:
class UItemShopNotificationsConfig_TA* ItemShopNotificationsConfig; // 0x0098 (0x0008) [0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ItemShopNotificationsConfig_TA");
}

return uClassPointer;
};

void HandleConfigChanged();

// Class TAGame.GFxData_KismetOverlay_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_KismetOverlay_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_KismetOverlay_TA");
}

return uClassPointer;
};

void DisableKismetOverlay(class FString UIOverlay);
void EnableKismetOverlay(class FString UIOverlay);
void ToggleOverlay(class FString UIOverlayName, unsigned long bToggleOn);
};

// Class TAGame.GFxData_LanBrowser_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_LanBrowser_TA : public UGFxDataSingleton_X
{
public:
class ULanBrowser_X*           LanBrowser;          // 0x0098 (0x0008)
[0x0000800000000000]
class UAsyncTask*               GetServerListTask; // 0x00A0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_LanBrowser_TA");
}

```

```

return uClassPointer;
};

void __GFxDLanBrowser_TA__Refresh_0x2(class UError* _);
void __GFxDLanBrowser_TA__Refresh_0x1(class UError* Error);
void CreateServerGFxDLanBrowser(class ULanServerRecord_X* Server);
void OnRefreshed();
void OnRefreshError();
void HandleServers(TArray<class ULanServerRecord_X*> Records);
void Refresh();
};

// Class TAGame.GFxDLanServer_TA
// 0x0084 (0x0094 - 0x0118)
class UGFxDLanServer_TA : public UGFxDLanServerRow_X
{
public:
class FString SanitizedOwnerName; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture* OwnerAvatar; // 0x00A8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString OwnerPlatform; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString LocalizedMapName; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString SanitizedServerName; // 0x00D0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ServerId; // 0x00E0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t ServerBuildID; // 0x00F0 (0x0004)
[0x0000000000000000]
class UOnlineGame_X* OnlineGame; // 0x00F8 (0x0008)
[0x0000800000000000]
class UOnlineSubsystem* OnlineSubsystem; // 0x0100 (0x0008)
[0x0000800000000000]
class UOnlineGameWorldFilter_X* WorldFilter; // 0x0108 (0x0008)
[0x0000800000000000]
class ULanBrowser_X* LanBrowser; // 0x0110 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxDLanServer_TA");
}

return uClassPointer;
};

```

```

void __GFxData_LanServer_TA__Connect_0x1(class UError* X);
void Connect(class FString Password);
class FString GetJoinOptions(class FString Password);
void HandleGetAvatar(struct FUniqueNetId InPlayerId, class UTexture* InAvatar, class FString OnlinePlayerName);
void LoadAvatar(struct FUniqueNetId PlayerID);
void HandleServerNameSanitized(class FString Original, class FString Sanitized);
void SanitizeServerName(class FString InServerName);
void HandlePlayerNameSanitized(class FString Original, class FString Sanitized);
void SanitizeOwnerName(struct FUniqueNetId InPlayerId, class FString InPlayerName);
class FString GetLocalizedMapName(struct FName MapID);
void SetData(class ULanServerRecord_X* Server);
};

// Class TAGame.GFxData_LanMatch_TA
// 0x0025 (0x0098 - 0x00BD)
class UGFxData_LanMatch_TA : public UGFxDataSingleton_X
{
public:
    class FString           ServerName;          // 0x0098 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString           Password;            // 0x00A8 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    int32_t                SelectedMaxPlayerCount; // 0x00B8 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    uint8_t                SettingsType;         // 0x00BC (0x0001)
    [0x0000000000000002] (CPF_Const)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_LanMatch_TA");
        }

        return uClassPointer;
    }

    void CreateMatch();
    void UpdateCustomMatchSettings();
    void SetPassword(class FString InPassword);
    void SetServerName(class FString InServerName);
    void SetSelectedMaxPlayerCount(int32_t MaxPlayerCount);
    class UProfile_TA* GetProfile();
    void SetCustomMatchSettings(struct FCustomMatchSettings& InSettings);
    struct FCustomMatchSettings GetCustomMatchSettings();
    void eventOnShellSet();
    class UOnlineGame_TA* GetOnlineGame();
};

// Class TAGame.LanServerMetaData_TA

```

```

// 0x0084 (0x0060 - 0x00E4)
class ULanServerMetaData_TA : public UObject
{
public:
    struct FUniqueNetId           OwnerId;          // 0x0060 (0x0048)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString                 OwnerName;        // 0x00A8 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    class FString                 ServerName;       // 0x00B8 (0x0010)
    [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
    struct FName                  ServerMap;        // 0x00C8 (0x0008)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t                      ServerGameMode;   // 0x00D0 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    unsigned long                bPassword : 1;    // 0x00D4 (0x0004)
    [0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
    int32_t                      NumPlayers;       // 0x00D8 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t                      MaxPlayers;       // 0x00DC (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t                      BuildID;          // 0x00E0 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.LanServerMetaData_TA");
        }

        return uClassPointer;
    };
};

// Class TAGame.GFxData_LastChanceBan_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_LastChanceBan_TA : public UGFxDataSingleton_X
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_LastChanceBan_TA");
        }

        return uClassPointer;
    };
};

```

```

return uClassPointer;
};

void __GFxData_LastChanceBan_TA_DismissLastChanceModal_0x1(class UBanSave_TA*
BanSave);
void OnShowLastChanceModal();
void DismissLastChanceModal();
static void HandleLastChance(class UBanSave_TA* InBanSave, class
UGFxData_LastChanceBan_TA* InGFxLastChance, class UPrimaryAuthLoggedIn_TA*
InLoggedIn);
void eventOnShellSet();
};

// Class TAGame.LeaderboardsConfig_TA
// 0x0010 (0x0078 - 0x0088)
class ULeaderboardsConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FLeaderboardRequestInfo>      RankedLeaderboards;          // 0x0078
(0x0010) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LeaderboardsConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_LegacyStatus_TA
// 0x0002 (0x0098 - 0x009A)
class UGFxData_LegacyStatus_TA : public UGFxDataSingleton_X
{
public:
uint8_t          LegacyStatus;           // 0x0098 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
uint8_t          LegacySetState;         // 0x0099 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_LegacyStatus_TA");
}

```

```

}

return uClassPointer;
};

void __GFxData_LegacyStatus_TA__HandleSaveDataLoaded_0x1(class
UOnlineStorageSyncManager_TA* _);
bool LegacyStatusSet();
void HandleLegacyOnlineSet();
void OnLegacyCheckComplete(class URPC_GetLegacyStatus_TA* RPC);
void HandlePsyNetConnection();
void HandleEpicError(class UEpicLogin_X* EpicLogin, class UErrorType* Error);
void HandleLoginResult(class UOnlinePlayerAuthentication_X* Auth);
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*
MySaveData, class UError* Error);
uint8_t GetSetState();
bool HasAllowableAuthError(class UOnlinePlayerAuthentication_X* Auth);
void eventOnShellSet();
};

// Class TAGame.RPC_GetLegacyStatus_TA
// 0x0019 (0x00E8 - 0x0101)
class URPC_GetLegacyStatus_TA : public URPC_X
{
public:
int32_t LegacyStatus; // 0x00E8 (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<struct FOnlineProductData> Products; // 0x00F0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
uint8_t ProcessedLegacyStatus; // 0x0100 (0x0001)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetLegacyStatus_TA");
}

return uClassPointer;
};

void eventOnSuccess();
};

// Class TAGame.GFxData_LegalText_TA
// 0x0010 (0x0098 - 0x0A8)
class UGFxData_LegalText_TA : public UGFxDataSingleton_X
{
public:
TArray<uint8_t> LegacyLegalPlatforms; // 0x0098 (0x0010)
}

```

[0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_LegalText_TA");  
}  
  
return uClassPointer;  
};  
  
void __GFxData_LegalText_TA__HandleSaveDataLoaded_0x1(class  
UOnlineStorageSyncManager_TA* _);  
bool __GFxData_LegalText_TA__SyncLegal_0x1(uint8_t P);  
void NotifyPrivacyPolicyUpdated();  
void NotifyEulaUpdated();  
void SyncLegal();  
void HandleSyncSuccess();  
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*  
MySaveData, class UError* Error);  
void eventOnShellSet();  
};  
  
// Class TAGame.PrivacyPolicy_TA  
// 0x0008 (0x0090 - 0x0098)  
class UPrivacyPolicy_TA : public ULegalText_TA  
{  
public:  
class UPrivacyPolicyConfig_TA*           Config;          // 0x0090 (0x0008)  
[0x0000800000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.PrivacyPolicy_TA");  
}  
  
return uClassPointer;  
};  
  
void __PrivacyPolicy_TA__Construct_0x1(class UOnlineGame_X* OnlineGame);  
void __PrivacyPolicy_TA__Construct_0x2(class UOnlineLegalText_X* OnlineLegalText);  
void Accept();  
bool RequiresAcceptance();  
class UPrivacyPolicySave_TA* GetPrivacySave();  
void HandleConfig();
```

```

void eventConstruct();
};

// Class TAGame.GFxData_LoadoutSets_TA
// 0x00C8 (0x0098 - 0x0160)
class UGFxData_LoadoutSets_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGFxLoadoutSet> LoadoutSets; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UProfile_TA* Profile; // 0x00A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FLoadoutData RandomLoadouts[0x2]; // 0x00B0 (0x0080)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UProductLoader_TA* RandomLoadoutLoader; // 0x0130
(0x0008) [0x0000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UProductLoader_TA* PreviousLoadoutSetLoader; // 0x0138
(0x0008) [0x0000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class UProductLoader_TA* NextLoadoutSetLoader; // 0x0140
(0x0008) [0x0000000004080009] (CPF>Edit | CPF_ExportObject | CPF_Component | CPF>EditInline)
class ULoadoutSequencer_TA* LoadoutSequencer; // 0x0148
(0x0008) [0x0000000000002000] (CPF_Transient)
int32_t PreviousBodyProductID; // 0x0150 (0x0004)
[0x0000000000002000] (CPF_Transient)
class UProfileLoadoutSave_TA* LoadoutSave; // 0x0158 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_LoadoutSets_TA");
}

return uClassPointer;
};

bool IsInGarageMenu();
class ACarPreviewActor_TA* GetCarPreview();
void FinishRandomization();
void UpdatePreviewLoadout();
int32_t GetPreviewTeam();
void SetPreviewTeam(int32_t Team);
void SetEditingLoadoutSet(int32_t Index);
void RandomizeComplete();
void HandleLoadoutSequenceComplete(class ULoadoutSequencer_TA* Sequencer);
void RandomizePreset(int32_t Index);
void SetupNextRandomLoadoutSlot(int32_t SlotIndex, struct FLoadoutData& Loadouts);

```

```

void SetupNextRandomLoadout(struct FLoadoutData& Loadouts);
bool HasPresetNamed(class FString NewName);
void RenamePreset(int32_t Index, class FString NewName);
void DeletePreset(int32_t Index);
void CreatePreset();
bool CanCreatePreset();
void EquipPreset(int32_t Index);
void HandleEquippedLoadoutSetsChanged(class ULoadout_TA* InLoadout);
void ClearLoaderCache(class UProductLoader_TA* Loader);
void HandleLoadoutLoaded(class UProductLoader_TA* Loader);
void PreloadLoadout(struct FLoadoutData Loadout, class UProductLoader_TA* Loader);
void PreloadNextPrevLoadoutSets(int32_t Index);
void InitLoadoutSets();
void OnEventProductArchived(struct FProductHashID HashID);
void HandleArchiveSaveLoaded(class UProductsArchiveSave_TA* InArchiveSave);
void HandleLoadoutSaveLoaded(class UProfileLoadoutSave_TA* InLoadoutSave);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.LoadoutSequencer_TA
// 0x00B8 (0x0060 - 0x0118)
class ULoadoutSequencer_TA : public UObject
{
public:
TArray<struct FSequenceData> SlotSequence; // 0x0060 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long bDebug : 1; // 0x0070 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class ACarPreviewActor_TA* PreviewActor; // 0x0078 (0x0008)
[0x0000004000000000]
struct FLoadoutData FinalLoadout; // 0x0080 (0x0040)
[0x0000004000400000] (CPF_NeedCtorLink)
struct FLoadoutData CurrentLoadout; // 0x00C0 (0x0040)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventSequenceComplete__Delegate; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutSequencer_TA");
}

return uClassPointer;
};

void Finish();
class AActor* GetTimerObject();
void QueueNextSlot(float Delay);

```

```

void EquipNextSlot();
void OnSequenceComplete();
void HandleProductLoaded(struct FAssetLoadResult Result);
void QueueSlot(class UProductSlot_TA* Slot, float AnimDelay);
void Start(class ACarPreviewActor_TA* InPreviewActor, struct FScriptDelegate Callback, struct FLoadoutData& InLoadout);
void SequencerLog(class FString LogStr);
void EventSequenceComplete(class ULoadoutSequencer_TA* Sequencer);
};

// Class TAGame.LoadoutRandomizer_TA
// 0x0000 (0x0060 - 0x0060)
class ULoadoutRandomizer_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutRandomizer_TA");
}

return uClassPointer;
};

static int32_t GetRandomColorID(class UCarColorSet_TA* Set);
static void RandomizeTeamPaint(struct FLoadoutData& Data);
static void RandomizeSlot(class UProfile_TA* Profile, int32_t SlotIndex, struct FLoadoutData& Data);
static void RandomizeSlotFromProductList(class UProductSlot_TA* Slot, struct FLoadoutData& Data, TArray<class UProduct_TA*>& ProductList);
static void Randomize(class UProfile_TA* Profile, struct FLoadoutData& Loadouts);
static void RandomizeLoadout(class UProfile_TA* Profile, struct FLoadoutData& Loadout);
static void RandomizerLog(class FString LogStr);
};

// Class TAGame.GFxData_LocalCar_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_LocalCar_TA : public UGFxDataSingleton_X
{
public:
int32_t          Boost;           // 0x0098 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long      bOnBack : 1;       // 0x009C (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long      bViewingCarCam : 1; // 0x009C (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long      bShowItemsDisplay : 1; // 0x009C (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long      bShowTargetReticle : 1; // 0x009C (0x0004)

```

```

[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long          bItemActive : 1;           // 0x009C (0x0004)
[0x0000000040000000] [0x00000010] (CPF_EditInlineNotify)
int32_t                TimeTillItem;           // 0x00A0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                MaxTimeTillItem;        // 0x00A4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString           ItemName;              // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture2D*       ItemIcon;              // 0x00B8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
float                  ItemTimeRemaining;       // 0x00C0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                ItemTargetID;           // 0x00C4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_LocalCar_TA");
}

return uClassPointer;
};

void SetCycleTargetScreenPosition(float ScreenX, float ScreenY, unsigned long bOffScreen,
int32_t Direction);
void SetItemTargetScreenPosition(float ScreenX, float ScreenY, unsigned long bOffScreen);
void SetOnBack(unsigned long blsOnBack);
void UpdateViewingCarCam(class ACamera_X* Camera);
void HandleCameraStateChanged(class ACamera_X* Camera, class UCameraState_X*
CameraState);
void HandleCameraTargetChanged(class ACamera_TA* Camera, class AActor* Target);
void HandleCameraChanged(class APlayerController_X* PlayerController);
void BoostPercentageChanged(int32_t NewPercentage, unsigned long bAnimate);
void SetBoostPercent(int32_t NewPercentage, unsigned long bAnimateChange);
void eventOnShellSet();
};

// Class TAGame.GFxData_PlayerVanity_TA
// 0x0018 (0x0098 - 0x00B0)
class UGFxData_PlayerVanity_TA : public UGFxDataSingleton_X
{
public:
class UVanitySetManager_TA*      VanityMgr;           // 0x0098 (0x0008)
[0x0000800000002000] (CPF_Transient)
class UGFxData_PlayerAvatarBorder_TA*   GFxPlayerAvatarBorder;    // 0x00A0
(0x0008) [0x0000004000002000] (CPF_Transient)
class UGFxData_PlayerBanner_TA*      GFxPlayerBanner;      // 0x00A8
(0x0008) [0x0000004000002000] (CPF_Transient)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerVanity_TA");
}

return uClassPointer;
};

void __GFxData_PlayerVanity_TA__InitVanityItems_0x1(class UPlayer* P);
bool __GFxData_PlayerVanity_TA__UpdatePlayerAvatars_0x1(struct FUniqueNetId PID);
int32_t GetPlayerBorderIndex(struct FUniqueNetId PlayerID);
void OnPlayerAvatarBordersUpdated(TArray<struct FUniqueNetId> PlayerIds);
void UpdatePlayerAvatarBorders(TArray<struct FUniqueNetId> PlayerIds);
void InitPlayerAvatarBorder();
int32_t GetPlayerNameBannerIndex(class FString PlayerName);
int32_t GetPlayerBannerIndex(struct FUniqueNetId PlayerID);
void OnPlayerBannersUpdated(TArray<struct FUniqueNetId> PlayerIds);
void UpdatePlayerBanners(TArray<struct FUniqueNetId> PlayerIds);
void InitPlayerBanner();
int32_t GetPlayerNameAvatarIndex(class FString PlayerName);
int32_t GetPlayerAvatarIndex(struct FUniqueNetId PlayerID);
void OnPlayerAvatarsUpdated(TArray<struct FUniqueNetId> PlayerIds);
void UpdatePlayerAvatars(TArray<struct FUniqueNetId> PlayerIds);
void HandleLoginChanged(class UOnlinePlayerAuthentication_X* Auth);
void InitVanityItems();
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.ProfileGameplaySave_TA
// 0x0007 (0x00C8 - 0x00CF)
class UProfileGameplaySave_TA : public USaveObject_TA
{
public:
unsigned long          bMetric : 1;           // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bTeamColoredBoostMeter : 1;    // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long          bDisableDivisions : 1;        // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000004]
unsigned long          bDisableDivisionsSports : 1;   // 0x00C8 (0x0004)
[0x0000000000000000] [0x00000008]
uint8_t                HUDMessageThreshold;        // 0x00CC (0x0001)
[0x0000000000000000]
uint8_t                RankInfoDisplayType;        // 0x00CD (0x0001)
[0x0000000000000000]
uint8_t                NameplateMode;            // 0x00CE (0x0001)
[0x0000000000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProfileGameplaySave_TA");
}

return uClassPointer;
};

void __ProfileGameplaySave_TA__GetVersionDelegates_0x1(class UObject* SaveObj);
static bool GetMetricDefaultValue();
void GetVersionDelegates(TArray<struct FScriptDelegate>& VersionDelegates);
void OnCreate();
};

// Class TAGame.GFxData_TeamInfo_TA
// 0x0038 (0x0094 - 0x00CC)
class UGFxData_TeamInfo_TA : public UGFxDataRow_X
{
public:
class FString TeamName; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FColor TeamColor; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Score; // 0x00AC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UTexture* TeamLogo; // 0x00B0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t SeriesScore; // 0x00B8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class ATeam_TA* Team; // 0x00C0 (0x0008)
[0x0000000000002000] (CPF_Transient)
int32_t Difficulty; // 0x00C8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TeamInfo_TA");
}

return uClassPointer;
};

void UpdateDifficulty();

```

```

void HandleGameScoreUpdated(class ATeam_Soccar_TA* InTeam);
void HandleScoreUpdated(class ATeam_TA* InTeam);
void HandleColorsChanged(class ATeam_TA* InTeam);
void HandleLogoChanged(class ATeam_TA* InTeam);
void HandleNameChanged(class UTeamNameComponent_TA* InTeamName, class FString
NewName);
void SetTeam(class ATeam_TA* InTeam);
};

// Class TAGame.SkillDecayedEvent_TA
// 0x0000 (0x0060 - 0x0060)
class USkillDecayedEvent_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkillDecayedEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Map_TA
// 0x00BF8 (0x0098 - 0x01590)
class UGFxData_Map_TA : public UGFxDataSingleton_X
{
public:
struct FGFxMapData           RandomMapData;          // 0x0098 (0x0038)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FGFxMapData           RandomStandardMapData; // 0x00D0
(0x0038) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FGFxMapData           RandomRocketLabsMedleyMapData; // 0x0108
(0x0038) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FGFxMapData>    Data;                // 0x01408 (0x0010)
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                      ActiveMapIndex;        // 0x011850 (0x0004)
[0x0000000040000001] (CPF_Edit | CPF_EditInlineNotify)
class UTexture2D*             LastLoadedMapImage;   // 0x012058
(0x0008) [0x0000000000000000]
struct FName                  RandomMapName;        // 0x012860 (0x0008)
[0x0000000000000002] (CPF_Const)
struct FName                  RandomStandardMapName; // 0x01368
(0x0008) [0x0000000000000002] (CPF_Const)
struct FName                  RandomRocketLabsMedleyMapName; // 0x0170
(0x0008) [0x0000000000000002] (CPF_Const)
struct FScriptDelegate         __EventMapListChanged__Delegate; // 0x01378

```

(0x0018) [0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Map_TA");  
}  
  
return uClassPointer;  
};  
  
void OnLargeMapImageFailed(struct FName MapName);  
void OnLargeMapImageLoaded(class UTexture2D* Image, struct FName MapName);  
class UGameMode_TA* GetGameModeFromMapDataName(struct FName MapDataName);  
void HandleLargeMapImageLoaded(struct FMapImageLoadResult Result);  
void LoadLargeMapImage(struct FName MapName);  
int32_t GetMapIndex(struct FName MapName);  
struct FGFxMapData CreateMapData(class UMapData_TA* Map);  
void SetMaps(TArray<struct FGFxMapData>& InData);  
void UpdateMapList(int32_t GameModeIndex, unsigned long AddRandomOption);  
int32_t SortFreeplayMaps(struct FGFxMapData A, struct FGFxMapData B);  
void UpdateFreeplayMapList();  
void eventOnShellSet();  
void EventMapListChanged(class UGFxData_Map_TA* Maps);  
};  
  
// Class TAGame.GFxData_MapPrefs_TA  
// 0x0058 (0x0098 - 0x00F0)  
class UGFxData_MapPrefs_TA : public UGFxDataSingleton_X  
{  
public:  
TArray<struct FGFxPlaylistType> PlaylistTypes; // 0x0098 (0x0010)  
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)  
TArray<struct FGFxMapType> MapTypes; // 0x00A8 (0x0010)  
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)  
TArray<struct FGFxMapPlaylist> Playlists; // 0x00B8 (0x0010)  
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)  
TArray<struct FGFxMapPreference> Maps; // 0x00C8 (0x0010)  
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)  
class UMapPrefsConfig_X* MapPrefsConfig; // 0x00D8 (0x0008)  
[0x0000800000000001] (CPF_Edit)  
class UMapPrefsSave_TA* MapPrefsSave; // 0x00E0 (0x0008)  
[0x0000000000002000] (CPF_Transient)  
class UMapsConfig_TA* MapsConfig; // 0x00E8 (0x0008)  
[0x0000800000002000] (CPF_Transient)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MapPrefs_TA");
}

return uClassPointer;
};

void __GFxData_MapPrefs_TA__OnShellSet_0x1(class UMapPrefsSave_TA* Save);
TArray<class UMapData_TA*> __GFxData_MapPrefs_TA__InitMapPrefs_0x2(class
UGameSettingPlaylist_X* Playlist);
bool __GFxData_MapPrefs_TA__InitMapPrefs_0x1(class UGameSettingPlaylist_X* Playlist);
class UProfile_TA* GetProfile();
static TArray<class UMapData_TA*> FilterAndSortMaps(TArray<class UMapData_TA*>
UnsortedMaps);
void SavePrefs(struct FPlaylistMapPrefs Prefs);
void SetMapPreference(struct FName PlaylistId, struct FName MapID, uint8_t Preference);
void SetPlaylistOverrideGlobal(struct FName PlaylistId, unsigned long bOverride);
struct FName GetMapType(class UMapData_TA* Map);
void AddPlaylistMapPrefs(struct FName PlaylistId, TArray<class UMapData_TA*> PlaylistMaps);
void AddPlaylistData(struct FName PlaylistId, struct FName PlaylistTypeID, class FString
PlaylistLabel, TArray<class UMapData_TA*> PlaylistMaps);
TArray<class UMapData_TA*> GetGlobalMaps();
class FString GetGlobalLabel();
void InitMapPrefs();
void HandleMapsConfigChange();
void InitMapTypes();
void InitPlaylistTypes();
void eventOnShellSet();
};

// Class TAGame.GFxData_MapSelection_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_MapSelection_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGfxMapSelectionData> Selections; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FScriptDelegate _EventMapListChange__Delegate; // 0x00A8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate _EventSelectionChange__Delegate; // 0x00C0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MapSelection_TA");
}

```

```

return uClassPointer;
};

struct FGFxMapSelectionData
__GFxData_MapSelection_TA_HandleMapListChanged_0x1(struct FGFxMapData M);
struct FName __GFxData_MapSelection_TA_GetDisabledMaps_0x2(struct
FGFxMapSelectionData MapItem);
bool __GFxData_MapSelection_TA_GetDisabledMaps_0x1(struct FGFxMapSelectionData
MapItem);
void SetSelected(int32_t Row, unsigned long bSelected);
void SetDisabledMaps(TArray<struct FName>& MapNames);
TArray<struct FName> GetDisabledMaps();
void HandleMapListChanged(class UGFxData_Map_TA* GFxMaps);
void eventOnShellSet();
void EventSelectionChange(class UGFxData_MapSelection_TA* GFxMapSelection, struct FName
MapName, unsigned long bSelected);
void EventMapListChange(class UGFxData_MapSelection_TA* GFxMapSelection);
};

// Class TAGame.GFxData_MatchAdmin_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_MatchAdmin_TA : public UGFxDataSingleton_X
{
public:
unsigned long           bMatchAdmin : 1;           // 0x0098 (0x0004)
[0x0000000040002000] [0x00000001] (CPF_Transient | CPF_EditInlineNotify)
class APRI_TA*          PlayerPRI;             // 0x00A0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MatchAdmin_TA");
}

return uClassPointer;
};

void SetPaused(unsigned long bNewPause);
void SetScoreAndTime(int32_t NewScoreTeam0, int32_t NewScoreTeam1, int32_t
SecondsRemaining, unsigned long bOverTime, unsigned long bRestartRound);
void HandleMatchAdmin(class APRI_TA* PRI);
void HandlePRIReceived(class APlayerController_X* PC_X);
class APlayerController_TA* GetPlayerController();
void eventOnRemoved();
void OnShellSet();
};

// Class TAGame.GFxData_OnlineMatchStatus_TA
// 0x0080 (0x0098 - 0x0118)

```

```

class UGFxData_OnlineMatchStatus_TA : public UGFxDataSingleton_X
{
public:
unsigned long          bSearching : 1;           // 0x0098 (0x0004)
[0x0008004040002000] [0x00000001] (CPF_Transient | CPF_EditInlineNotify)
struct FName            SearchState;           // 0x009C (0x0008)
[0x0000004040002000] (CPF_Transient | CPF_EditInlineNotify)
class FString           SavedSearchingString;    // 0x00A8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FMatchmakingMessage>   SavedSearchStrings;      // 0x00B8
(0x0010) [0x0008004000402000] (CPF_Transient | CPF_NeedCtorLink)
uint8_t                 StatusOwner;           // 0x00C8 (0x0001)
[0x0000004040002000] (CPF_Transient | CPF_EditInlineNotify)
class UOnlineGameParty_X* OnlineParty;         // 0x00D0 (0x0008)
[0x0000004000002000] (CPF_Transient)
class FString           PartyLeaderSearchingMessage; // 0x00D8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FScriptDelegate   __bSearching__ChangeNotify; // 0x00E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate   __SavedSearchStrings__ChangeNotify; // 0x0100
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_OnlineMatchStatus_TA");
}

return uClassPointer;
};

void __SavedSearchStrings__ChangeNotifyFunc();
void __bSearching__ChangeNotifyFunc();
TArray<struct FMatchmakingMessage> GetSearchStringsArray();
class FString GetSearchStrings();
void SaveSearchString(class FString NewString);
void OnCountdownEnded();
void OnCountdownStarted(int32_t SecondsRemaining);
void OnClearMessages();
void OnSearchError(class FString Message);
void OnSearchMessage(class FString Message, uint8_t MessageType);
void OnFindGameComplete();
void ClearMessages();
void AddMessage(class FString InMessage, uint8_t InMessageType);
class UOnlineGame_X* GetOnlineGame();
void HandleCountdownEnded();
void HandleCountdownStarted();
void HandlePartyJoinGameError(class UOnlineGameParty_X* Party, class FString LocalizedErrorString);
void HandleFindGameWarning(class FString Message);

```

```

void SetError(class FString msg);
void SetWarning(class FString msg);
void SetStatus(class FString msg);
void HandleFindGameComplete(unsigned long bCancelled);
void HandleFindGameStatusChanged(class FString NewStatus);
void SetSearching(unsigned long bNewSearching, uint8_t NewOwner);
void SetSearchState(struct FName NewState);
void SetPartyLeaderSearchingMessage();
void HandlePartySizeChanged(class UOnlineGameParty_X* Party, int32_t NewSize, int32_t OldSize);
void HandleJoinGameComplete(unsigned long bSuccess, class FString FailReason);
void HandlePartySearchStatus(class UOnlineGameParty_X* Party, class UPartyMessage_SearchStatus_X* StatusMessage);
void SetStatusOwner(uint8_t NewOwner);
void eventOnShellSet();
};

// Class TAGame.GFxData_Regions_TA
// 0x0038 (0x0098 - 0x00D0)
class UGFxData_Regions_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FGfxRegion> SuperRegions; // 0x0098 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FGfxRegion> Regions; // 0x00A8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventRegionsSynced__Delegate; // 0x00B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Regions_TA");
}

return uClassPointer;
};

TArray<struct FGfxRegion> SelectRegions(TArray<class FString> SelectedRegions);
void NotifyWhenSynced(struct FScriptDelegate Callback);
void HandleRegionsSynced(class UOnlineGameRegions_X* RegionsObj);
void eventOnShellSet();
void EventRegionsSynced(class UGFxData_Regions_TA* RegionsData);
};

// Class TAGame.MatchmakingViewFilters_TA
// 0x0080 (0x0060 - 0x00E0)
class UMatchmakingViewFilters_TA : public UObject
{
public:

```

```

class UIMatchmakingViewFilter_TA*          Filters_Object[0x8];           // 0x0060
(0x0078) [0x0000000000000000]
class UIMatchmakingViewFilter_TA*          Filters_Interface[0x8];        // 0x0068
(0x0078) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilters_TA");
}

return uClassPointer;
};

TArray<struct FName> GetAccessiblePlaylists(uint8_t ViewTab, struct FName
LastVisitedPlaylistFolderID, TArray<struct FName>& SelectedPlaylists);
void eventConstruct();
};

// Class TAGame.MatchmakingSettingsSave_TA
// 0x002C (0x00C8 - 0x00F4)
class UMatchmakingSettingsSave_TA : public USaveObject_TA
{
public:
uint8_t                  MatchmakingViewTab;           // 0x00C8 (0x0001)
[0x0000000000000000] 
TArray<struct FName>      QuickMatchPlaylists;        // 0x00D0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class FString>    QuickMatchRegions;         // 0x00E0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long             bMultiSelectEnabled : 1;     // 0x00F0 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingSettingsSave_TA");
}

return uClassPointer;
};

// Class TAGame.ViewingMatchmaking_TA
// 0x0000 (0x0060 - 0x0060)

```

```

class UViewingMatchmaking_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ViewingMatchmaking_TA");
}

return uClassPointer;
};

};

// Class TAGame.MenuTreeConfig_TA
// 0x00084 (0x0078 - 0x00807C)
class UMenuTreeConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bEnableDuelingIcons : 1;           // 0x0078 (0x0004)
[0x0001000000000001] [0x00000001] (CPF_Edit)
int32_t                 PlaylistCacheDuration;           // 0x007C8 (0x0004)
[0x0001000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_MenuTreeNode_TA
// 0x0035 (0x0094 - 0x00C9)
class UGFxData_MenuTreeNode_TA : public UGFxDataRow_X
{
public:
unsigned long          bEnabled : 1;                   // 0x0098 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bDisplayCrumbTrail : 1;        // 0x0098 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
class UTexture*         ThumbnailImage;               // 0x00A0 (0x0008)

```

```
[0x0001000040000000] (CPF_EditInlineNotify)
class FString                                BannerTitle;           // 0x00A8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString                                BannerBody;            // 0x00B8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint8_t                                     BannerType;           // 0x00C8 (0x0001)
[0x0001000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MenuTreeNode_TA");
}

return uClassPointer;
};

void SetBanner(uint8_t InBannerType, class FString InBannerTitle, class FString InBannerBody);
bool CanShowEngagementEventType(uint8_t EventType);
void ClearNode();
void SetNode(class UMenuTreeNode_TA* InNode);
};

// Class TAGame.MenuTreeNode_GenericFolder_TA
// 0x0004 (0x01B0 - 0x01B4)
class UMenuTreeNode_GenericFolder_TA : public UMenuTreeBranch_TA
{
public:
unsigned long                                bCanShowQuickMatchButton : 1;           // 0x01B0
(0x0004) [0x0001000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_GenericFolder_TA");
}

return uClassPointer;
};

bool IsAbleToQuickSearch();
void UpdateQuickMatchButton();
void OnGFxNodeSet();
};

// Class TAGame.MenuTreeNode_PlaylistFolder_TA
```

```

// 0x004C4 (0x01B4 - 0x02001F8)
class UMenuTreeNode_PlaylistFolder_TA : public UMenuTreeNode_GenericFolder_TA
{
public:
unsigned long          bCompetitive : 1;           // 0x01B8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bShowArenasAndRegions : 1;    // 0x01B8 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long          bCanUseMultiSelect : 1;       // 0x01B8 (0x0004)
[0x0001000040000000] [0x00000004] (CPF_EditInlineNotify)
uint8_t                HeaderType;                 // 0x01BC (0x0001)
[0x0001000040000000] (CPF_EditInlineNotify)
uint64_t               HeaderEndTime;              // 0x01C0 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
class FString          LTMPlaylistName;            // 0x01C8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          DuelingPlaylistNameA;        // 0x01D8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString          DuelingPlaylistNameB;        // 0x01E8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

class UMenuTreeConfig_TA*      MenuTreeConfig;        // 0x01F8
(0x0008) [0x0001800000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_PlaylistFolder_TA");
}

return uClassPointer;
};

void __MenuTreeNode_PlaylistFolder_TA__OnGFxNodeSet_0x1(class
UMenuTreeNode_Playlist_TA* PlaylistNode);
void __MenuTreeNode_PlaylistFolder_TA__OnGFxNodeSet_0x2(class
UMenuTreeNode_Playlist_TA* _);
bool __MenuTreeNode_PlaylistFolder_TA__UpdateSpecialPlaylistInfo_0x3(class
UMenuTreeNode_Playlist_TA* N);
bool __MenuTreeNode_PlaylistFolder_TA__UpdateSpecialPlaylistInfo_0x2(class
UMenuTreeNode_Playlist_TA* N);
void __MenuTreeNode_PlaylistFolder_TA__UpdateSpecialPlaylistInfo_0x1(class
UMenuTreeNode_Playlist_TA* N);
void __MenuTreeNode_PlaylistFolder_TA__Cleanup_0x1(class UMenuTreeNode_Playlist_TA*
PlaylistNode);
void HandleOnlineMatchStatusChanged();
void Cleanup();
void OnClicked();
void HandlePlaylistNodeClicked(class UMenuTreeNode_Playlist_TA* PlaylistNode);

```

```
void UpdateSpecialPlaylistInfo();
void BatchHandleChildPlaylistsChanged();
void OnGFxNodeSet();
};

// Class TAGame.GFxData_SaveData_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_SaveData_TA : public UGFxDataSingleton_X
{
public:
class UUISavedValues_TA*           PersistentSavedValues;          // 0x0098
(0x0008) [0x0000000000002000] (CPF_Transient)
class UUISavedValues_TA*           TransientSavedValues;         // 0x00A0
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SaveData_TA");
}

return uClassPointer;
};

void __GFxData_SaveData_TA__OnShellSet_0x1(class UUISavedValues_TA* Values);
void SaveTransientValue(struct FName Key, class FString Value);
bool HasTransientValue(struct FName Key);
class FString GetTransientValue(struct FName Key);
void SaveGlobalValue(struct FName Key, class FString Value);
bool HasGlobalValue(struct FName Key);
class FString GetGlobalValue(struct FName Key);
void OnShellSet();
};

// Class TAGame.MenuTreeNodeEnabledChecker_TA
// 0x0000 (0x0060 - 0x0060)
class UMenuTreeNodeEnabledChecker_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNodeEnabledChecker_TA");
}
}
```

```
return uClassPointer;
};

static bool __MenuTreeNodeEnabledChecker_TA__UpdateGFxNodeEnabled_0x2(class UMenuTreeNode_TA* N);
static bool __MenuTreeNodeEnabledChecker_TA__UpdateGFxNodeEnabled_0x1(class UMenuTreeNode_TA* N);
static void UpdateGFxNodeEnabled(class UGFxData_MenuTreeNode_TA* InGFxNode, class UMenuTreeNode_TA* InOwningNode);
};

// Class TAGame.GFxData_MenuTreeNode_Custom_TA
// 0x0007 (0x00C9 - 0x00D0)
class UGFxData_MenuTreeNode_Custom_TA : public UGFxData_MenuTreeNode_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MenuTreeNode_Custom_TA");
}

return uClassPointer;
};

};

// Class TAGame.MenuTreeNode_Custom_TA
// 0x0000 (0x0178 - 0x0178)
class UMenuTreeNode_Custom_TA : public UMenuTreeNode_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_Custom_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_MenuTreeNode_ExternalScreen_TA
```

```
// 0x0007 (0x00C9 - 0x00D0)
class UGFxData_MenuTreeNode_ExternalScreen_TA : public UGFxData_MenuTreeNode_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_MenuTreeNode_ExternalScreen_TA");
}

return uClassPointer;
};

};

// Class TAGame.MenuTreeNode_ExternalScreen_TA
// 0x0010 (0x0178 - 0x0188)
class UMenuTreeNode_ExternalScreen_TA : public UMenuTreeNode_TA
{
public:
class FString           ScreenToOpen;          // 0x0178 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_ExternalScreen_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_MenuTreeNode_GenericFolder_TA
// 0x000B (0x00C9 - 0x00D4)
class UGFxData_MenuTreeNode_GenericFolder_TA : public UGFxData_MenuTreeNode_TA
{
public:
unsigned long           bShowQuickMatchButton : 1;      // 0x00D0 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_MenuTreeNode_GenericFolder_TA");
}

return uClassPointer;
};

void UpdateQuickMatchButton();
};

// Class TAGame.GFxData_MenuTreeNode_Playlist_TA
// 0x001F (0x00C9 - 0x00E8)
class UGFxData_MenuTreeNode_Playlist_TA : public UGFxData_MenuTreeNode_TA
{
public:
unsigned long          bSelected : 1;           // 0x00D0 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bLocked : 1;            // 0x00D0 (0x0004)
[0x0001000040000000] [0x00000002] (CPF_EditInlineNotify)
class FString          LockReason;           // 0x00D8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MenuTreeNode_Playlist_TA");
}

return uClassPointer;
};

uint64_t GetCurrentPlaylistEndTime();
void UpdateSelected(unsigned long bNewSelected);
void UpdateLock();
};

// Class TAGame.GFxData_MenuTreeNode_PlaylistFolder_TA
// 0x0008 (0x00D4 - 0x00DC)
class UGFxData_MenuTreeNode_PlaylistFolder_TA : public
UGFxData_MenuTreeNode_GenericFolder_TA
{
public:
unsigned long          bUsingMultiselect : 1;      // 0x00D8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_MenuTreeNode_PlaylistFolder_TA");
}

return uClassPointer;
};

bool __GFxData_MenuTreeNode_PlaylistFolder_TA__DeselectAllPlaylists_0x1(class
UMenuTreeNode_TA* N);
void DeselectAllPlaylists();
void ToggleMultiSelect();
};

// Class TAGame.GFxData_MenuTreeNode_PrivateMatchJoin_TA
// 0x0000 (0x00D0 - 0x00D0)
class UGFxData_MenuTreeNode_PrivateMatchJoin_TA : public
UGFxData_MenuTreeNode_Custom_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_MenuTreeNode_PrivateMatchJoin_TA");
}

return uClassPointer;
};

};

// Class TAGame.MenuTreeNode_PrivateMatchJoin_TA
// 0x0018 (0x0178 - 0x0190)
class UMenuTreeNode_PrivateMatchJoin_TA : public UMenuTreeNode_Custom_TA
{
public:
class UGFxData_Party_TA*           Party;           // 0x0178 (0x0008)
[0x0001000000000000]
class FString                      PartyMatchKey; // 0x0180 (0x0010)
[0x000100000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_PrivateMatchJoin_TA");
}

return uClassPointer;
};

void HandlePlayerInMatchChanged(class UOnlineGameParty_X* PartyObject, struct FUniqueNetId InMemberId);
void HandlePartyChanged(class UOnlineGameParty_X* PartyObject);
void UpdateTitle();
void Cleanup();
void OnGFxNodeSet();
};

// Class TAGame.GFxData_MenuTreeNode_Tournaments_TA
// 0x0000 (0x00D0 - 0x00D0)
class UGFxData_MenuTreeNode_Tournaments_TA : public
UGFxData_MenuTreeNode_Custom_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_MenuTreeNode_Tournaments_TA");
}

return uClassPointer;
};

void CacheTimeUntilNextTournament(int32_t AutoTourTourCardRow, unsigned long
bRegistered);
void UpdateBadge(int32_t AutoTourTourCardRow, unsigned long bRegistered);
void ClearTournament();
};

// Class TAGame.MenuTreeNode_Tournaments_TA
// 0x0008 (0x0178 - 0x0180)
class UMenuTreeNode_Tournaments_TA : public UMenuTreeNode_Custom_TA
{
public:
uint64_t NextTournamentStartTime; // 0x0178 (0x0008)
[0x0001000000000000]
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreeNode_Tournaments_TA");
}

return uClassPointer;
};

void ClearTimer();
void UpdateTimeUntilNextTournament();
void CacheTimeUntilNextTournament(int32_t NextTournamentRow, unsigned long bRegistered);
void UpdateBadge(int32_t NextTournamentRow, unsigned long bRegistered);
void ClearTournament();
};

// Class TAGame.GFxData_MicroEvent_TA
// 0x0014 (0x0098 - 0x00AC)
class UGFxData_MicroEvent_TA : public UGFxDataSingleton_X
{
public:
class UMicroEventConfig_TA*           MicroEventConfig;          // 0x0098
(0x0008) [0x0000800000000000]
class UMicroEvent_TA*                 MicroEvent;              // 0x00A0 (0x0008)
[0x0000000000000000]
unsigned long                         bIsActive : 1;          // 0x00A8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MicroEvent_TA");
}

return uClassPointer;
};

void UpdateActiveStatus();
void UpdateActiveEvent();
void HandleConfigChanged();
};

// Class TAGame.MicroEvent_TA
// 0x0018 (0x0060 - 0x0078)
class UMicroEvent_TA : public UObject

```

```

{
public:
int32_t PlaylistId; // 0x0060 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
uint64_t StartTime; // 0x0068 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
uint64_t EndTime; // 0x0070 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MicroEvent_TA");
}

return uClassPointer;
};

bool IsValid();
bool IsActive(uint64_t CurrentTime);
};

// Class TAGame.MicroEventConfig_TA
// 0x0020 (0x0078 - 0x0098)
class UMicroEventConfig_TA : public UOnlineConfig_X
{
public:
TArray<class UMicroEvent_TA*> MicroEvents; // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UIEpochNow* Time_Object; // 0x0088 (0x0008)
[0x0000000000000000]
class UIEpochNow* Time_Interface; // 0x0090 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MicroEventConfig_TA");
}

return uClassPointer;
};

class UMicroEvent_TA* __MicroEventConfig_TA__GetNextActiveMicroEvent_0x3(class UMicroEvent_TA* Best, class UMicroEvent_TA* Current);
bool __MicroEventConfig_TA__GetNextActiveMicroEvent_0x2(class UMicroEvent_TA* M);

```

```

bool __MicroEventConfig_TA__GetNextActiveMicroEvent_0x1(class UMicroEvent_TA* M);
bool IsActiveMicroEventPlaylist(int32_t PlaylistId);
class UMicroEvent_TA* GetNextActiveMicroEvent();
void Apply();
void eventConstruct();
};

// Class TAGame.GFxData_MiniScoreboard_TA
// 0x0014 (0x0098 - 0x00AC)
class UGFxData_MiniScoreboard_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FScoreIndex> Data; // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t MiniScoreboardSlots; // 0x00A8 (0x0004)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MiniScoreboard_TA");
}

return uClassPointer;
};

bool __GFxData_MiniScoreboard_TA__Update_0x1(class APRI_TA* PRI);
int32_t GetSuccessMeasure(class AGameEvent_KnockOut_TA* GameEvent, class APRI_KnockOut_TA* PRI);
void Update(class AGameEvent_KnockOut_TA* GameEvent);
};

// Class TAGame.RPC_MicroTransactions_UnlockContainer_TA
// 0x0078 (0x00E8 - 0x0160)
class URPC_MicroTransactions_UnlockContainer_TA : public URPC_X
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceId> InstanceIDs; // 0x0130 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceId> KeyInstanceIDs; // 0x0140
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData> Drops; // 0x0150 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_MicroTransactions_UnlockContainer_TA");
}

return uClassPointer;
};

class URPC_MicroTransactions_UnlockContainer_TA* SetPlayerID(struct FUniqueNetId
InPlayerId);
class URPC_MicroTransactions_UnlockContainer_TA* SetKeyIDs(TArray<struct
FProductInstanceId>& InKeyIDs);
class URPC_MicroTransactions_UnlockContainer_TA* SetContainerIDs(TArray<struct
FProductInstanceId>& InContainerIDs);
};

// Class TAGame.RPC_MicroTransactions_GetCatalog_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_MicroTransactions_GetCatalog_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;                      // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FName                   Category;                     // 0x0130 (0x0008)
[0x0000000000000000]
TArray<struct FMTCatalogInfo> MTXProducts;                // 0x0138 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                 Currency;                    // 0x0148 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_MicroTransactions_GetCatalog_TA");
}

return uClassPointer;
};

static struct FName CategoryToName(uint8_t InCategory);
class URPC_MicroTransactions_GetCatalog_TA* SetCategory(uint8_t InCategory);
class URPC_MicroTransactions_GetCatalog_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_MicroTransactions_StartPurchase_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_MicroTransactions_StartPurchase_TA : public URPC_X
{

```

```

public:
class FString Language; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FUniqueNetId PlayerID; // 0x00F8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FMTCartItem> CartItems; // 0x0140 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_MicroTransactions_StartPurchase_TA");
}

return uClassPointer;
};

class URPC_MicroTransactions_StartPurchase_TA* SetLanguage(class FString InLanguage);
class URPC_MicroTransactions_StartPurchase_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
class URPC_MicroTransactions_StartPurchase_TA* SetCartItems(TArray<struct FMTCartItem>
InCartItems);
};

// Class TAGame.RPC_MicroTransactions_FinalizePurchase_TA
// 0x0088 (0x00E8 - 0x0170)
class URPC_MicroTransactions_FinalizePurchase_TA : public URPC_X
{
public:
uint64_t OrderId; // 0x00E8 (0x0008)
[0x0000000000000000]
unsigned long Authorized : 1; // 0x00F0 (0x0004)
[0x0000000000000000] [0x00000001]
struct FUniqueNetId PlayerID; // 0x00F8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData> Products; // 0x0140 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCurrency> Currencies; // 0x0150 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCurrency> WalletCurrencies; // 0x0160 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class

```

```

TAGame.RPC_MicroTransactions_FinalizePurchase_TA');
}

return uClassPointer;
};

class URPC_MicroTransactions_FinalizePurchase_TA* SetPlayerID(struct FUniqueNetId
InPlayerId);
class URPC_MicroTransactions_FinalizePurchase_TA* SetAuthorized(unsigned long
bAuthorized);
class URPC_MicroTransactions_FinalizePurchase_TA* SetOrderID(uint64_t InOrderID);
};

// Class TAGame.SeqEvent_MTXGarageBlueprintPreview_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_MTXGarageBlueprintPreview_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_MTXGarageBlueprintPreview_TA");
}

return uClassPointer;
};

void SetRevealRarity(uint8_t InRarity);
};

// Class TAGame.MultiltemDropConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UMultiltemDropConfig_TA : public UOnlineConfig_X
{
public:
unsigned long          bDropChallengeRewardsAsGroup : 1;           // 0x0078
(0x0004) [0x0000000000000001] [0x00000001] (CPF_Edit)
class UTexture*        CrewRocketPassDropIcon;                  // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MultiltemDropConfig_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.RewardDropGroup_TA
// 0x0020 (0x00C0 - 0x00E0)
class URewardDropGroup_TA : public UItemDropGroup_TA
{
public:
TArray<struct FGFxRewardDrop> Rewards; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FGFxLevelThreshold> Thresholds; // 0x00D0
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RewardDropGroup_TA");
}

return uClassPointer;
};

struct FGFxLevelThreshold __RewardDropGroup_TA__UpdateLevelThresholds_0x1(int32_t
Threshold);
bool IsValid();
void UpdateRewardInfo(struct FOnlineXPReward& InRewardDrop);
void UpdateLevelThresholds(struct FOnlineXPReward& InRewardDrop);
int32_t SortThresholdsAscending(struct FGFxLevelThreshold A, struct FGFxLevelThreshold B);
int32_t GetIndex(struct FOnlineXPReward& InRewardDrop);
void AddReward(struct FOnlineXPReward& InRewardDrop);
};

// Class TAGame.GFxData_PlayerTitles_TA
// 0x0028 (0x0098 - 0x00C0)
class UGFxData_PlayerTitles_TA : public UGFxDataSingleton_X
{
public:
int32_t SelectedTitle; // 0x0098 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
TArray<struct FPlayerTitleData> PlayerTitles; // 0x00A0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UProfileLoadoutSave_TA* LoadoutSave; // 0x00B0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UPlayerTitleConfig_X* TitleConfig; // 0x00B8 (0x0008)
[0x0000800000000000]

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerTitles_TA");
}

return uClassPointer;
};

void __GFxData_PlayerTitles_TA__HandleSaveDataLoaded_0x1(class USaveData_TA* SaveData);
void __GFxData_PlayerTitles_TA__HandleLoadoutSaveLoaded_0x1();
void __GFxData_PlayerTitles_TA__UpdatePlayerTitles_0x1(class UPlayer* P);
bool __GFxData_PlayerTitles_TA__GetOnlinePlayerTitles_0x3(struct FPlayerTitleData Title);
bool __GFxData_PlayerTitles_TA__GetOnlinePlayerTitles_0x2(struct FName Title);
struct FName __GFxData_PlayerTitles_TA__GetOnlinePlayerTitles_0x1(class UOnlineProduct_TA*
Product);
int32_t SortTitles(struct FPlayerTitleData A, struct FPlayerTitleData B);
TArray<struct FPlayerTitleData> GetOnlinePlayerTitles();
TArray<struct FPlayerTitleData> GetPlayerTitles();
struct FPlayerTitleData GetTitleData(struct FName TitleId);
void SetPlayerTitle(struct FName Title);
void UpdateSelectedTitle(struct FName Title);
void UpdatePlayerTitles();
void HandleNewOnlineProduct(class USaveData_TA* SaveData, class UOnlineProduct_TA*
OnlineProduct, class FString Message);
void HandleLoadoutSaveLoaded(class UProfileLoadoutSave_TA* InSave);
void HandleProfileSet(class ULocalPlayer_TA* P);
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA*
InSaveData, class UError* Error);
void eventOnShellSet();
};

// Class TAGame.MusicTheme_TA
// 0x0060 (0x0060 - 0x00C0)
class UMusicTheme_TA : public UObject
{
public:
TArray<class UMusicPlaylist_TA*> Playlists; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FPlaylistTitleTrack> TitleTracks; // 0x0070 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UAkSoundCue* NeverPlayInFirstTracklist; // 0x0080 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FPlaylistTrack Credits; // 0x0088 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FPlaylistTrack KonamiCode; // 0x0098 (0x0010)
[0x0000000000000001] (CPF_Edit)
class UTexture* DefaultIcon; // 0x00A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FPlaylistTrack> AdditionalTracks; // 0x00B0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTheme_TA");
}

return uClassPointer;
};

class UMusicTheme_TA* ApplyIcons();
};

// Class TAGame.MusicTracklistBuildStep_RemoveTrack_TA
// 0x0018 (0x0060 - 0x0078)
class UMusicTracklistBuildStep_RemoveTrack_TA : public UObject
{
public:
class UIMusicTracklistBuilder*           Source_Object;          // 0x0060 (0x0008)
[0x0000000000000000]
class UIMusicTracklistBuilder*           Source_Interface;     // 0x0068 (0x0008)
[0x0000000000000000]
class UAkSoundCue*                      TrackToRemove;        // 0x0070 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTracklistBuildStep_RemoveTrack_TA");
}

return uClassPointer;
};

class UMusicTracklist_TA* CreateMusicTracklist(TArray<class UMusicTrack_TA*> Tracks);
};

// Class TAGame. IMusicTracklist_TA
// 0x0000 (0x0060 - 0x0060)
class UIMusicTracklist_TA : public IInterface
{
public:

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IMusicTracklist_TA");
}

return uClassPointer;
};

class UIMusicTracklist_TA* Rebuild(class UMusicTracklistFactory_TA* Builder);
int32_t Num();
void NextTrack();
};

// Class TAGame.MusicUrlConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UMusicUrlConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FMusicConfigTrack>           Tracks;                                // 0x0078 (0x0010)
[0x0000000000040001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicUrlConfig_TA");
}

return uClassPointer;
};

class FString GetTrackUrl(struct FName TrackName);
};

// Class TAGame.MusicTracklistBuildStep_ShuffleTracks_TA
// 0x0010 (0x0060 - 0x0070)
class UMusicTracklistBuildStep_ShuffleTracks_TA : public UObject
{
public:
class UIMusicTracklistBuilder*           Source_Object;                           // 0x0060 (0x0008)
[0x0000000000000000]
class UIMusicTracklistBuilder*           Source_Interface;                      // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTracklistBuildStep_ShuffleTracks_TA");
}

return uClassPointer;
};

class UMusicTracklist_TA* CreateMusicTracklist(TArray<class UMusicTrack_TA*> Tracks);

// Class TAGame.MusicTracklistBuildStep_RemoveStreamerUnsafeTracks_TA
// 0x0018 (0x0060 - 0x0078)
class UMusicTracklistBuildStep_RemoveStreamerUnsafeTracks_TA : public UObject
{
public:
class UIMusicTracklistBuilder*           Source_Object;          // 0x0060 (0x0008)
[0x0000000000000000]
class UIMusicTracklistBuilder*           Source_Interface;     // 0x0068 (0x0008)
[0x0000000000000000]
class UStreamerSafeConfig_TA*           Config;              // 0x0070 (0x0008)
[0x0001800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.MusicTracklistBuildStep_RemoveStreamerUnsafeTracks_TA");
}

return uClassPointer;
};

bool
__MusicTracklistBuildStep_RemoveStreamerUnsafeTracks_TA__CreateMusicTracklist_0x1(class
UMusicTrack_TA* Track);
class UMusicTracklist_TA* CreateMusicTracklist(TArray<class UMusicTrack_TA*> Tracks);
};

// Class TAGame.MusicThemeConfig_TA
// 0x0008 (0x0078 - 0x0080)
class UMusicThemeConfig_TA : public UOnlineConfig_X
{
public:
class UMusicTheme_TA*                  Theme;                // 0x0078 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicThemeConfig_TA");
}

return uClassPointer;
};

void Apply();
};

// Class TAGame.MusicTracklistBuildStep_CreateDefault_TA
// 0x0000 (0x0060 - 0x0060)
class UMusicTracklistBuildStep_CreateDefault_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTracklistBuildStep_CreateDefault_TA");
}

return uClassPointer;
};

class UMusicTracklist_TA* CreateMusicTracklist(TArray<class UMusicTrack_TA*> Tracks);

// Class TAGame.MusicMetrics_TA
// 0x0014 (0x0080 - 0x0094)
class UMusicMetrics_TA : public UMetricsGroup_X
{
public:
struct FName TrackName; // 0x0080 (0x0008)
[0x0000000000000000] (CPF_Transient)
float TrackStartTime; // 0x0088 (0x0004)
[0x0000000000000000] (CPF_Transient)
int32_t TrackSequence; // 0x008C (0x0004)
[0x0000000000000000] (CPF_Transient)
float PendingMusicVolume; // 0x0090 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicMetrics_TA");
}

return uClassPointer;
};

float GetMusicVolume();
void PlaylistSelected(struct FName Playlist, unsigned long bSelected);
void TrackSkip(struct FName Track, int32_t Sequence, float Duration, float Volume);
void RecordTrackSkip();
void TrackEnd(struct FName Track, int32_t Sequence, float Volume);
void RecordTrackEnd();
void TrackStart(struct FName Track, int32_t Sequence, float Volume);
void RecordTrackStart(class UAkSoundCue* Cue);
};

// Class TAGame.GFxData_MusicStingers_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_MusicStingers_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_MusicStingers_TA");
}

return uClassPointer;
};

void ShowToast(struct FProductHashID HashID);
void UpdateStingerHashID(class UProductAsset_MusicStingers_TA* Stinger);
void HandleArenaSoundManagerSet(class UArenaSoundManager_TA* ArenaSoundManager);
void eventOnShellSet();
};

// Class TAGame.MutatorGameSettingPreset_TA
// 0x000C (0x0064 - 0x0070)
class UMutatorGameSettingPreset_TA : public UGameSetting_X
{
public:
class UPresetMutators_X* Presets; // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MutatorGameSettingPreset_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Notification_AchievementProgress_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_Notification_AchievementProgress_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_Notification_AchievementProgress_TA");
}

return uClassPointer;
};

};

// Class TAGame.Notification_AchievementProgress_TA
// 0x0018 (0x0170 - 0x0188)
class UNotification_AchievementProgress_TA : public UNotification_TA
{
public:
int32_t           Id;                      // 0x0170 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UTexture*    Icon;                     // 0x0178 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
float             Progress;                 // 0x0180 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float             MaxProgress;            // 0x0184 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.Notification_AchievementProgress_TA");
}

return uClassPointer;
};

class UNotification_TA* SetTitle(class FString InTitle);
class UNotification_AchievementProgress_TA* SetMaxProgress(float InMaxProgress);
class UNotification_AchievementProgress_TA* SetProgress(float InProgress);
class UNotification_AchievementProgress_TA* SetIcon(class UTexture* InIcon);
class UNotification_AchievementProgress_TA* SetID(int32_t InID);
};

// Class TAGame.GFxData_Notification_AchievementUnlocked_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_Notification_AchievementUnlocked_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_Notification_AchievementUnlocked_TA");
}

return uClassPointer;
};

};

// Class TAGame.Notification_AchievementUnlocked_TA
// 0x0010 (0x0170 - 0x0180)
class UNotification_AchievementUnlocked_TA : public UNotification_TA
{
public:
int32_t Id; // 0x0170 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Points; // 0x0174 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UTexture* Icon; // 0x0178 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.Notification_AchievementUnlocked_TA");
}

return uClassPointer;
};

class UNotification_TA* SetTitle(class FString InTitle);
class UNotification_AchievementUnlocked_TA* SetIcon(class UTexture* InIcon);
class UNotification_AchievementUnlocked_TA* SetPoints(int32_t InPoints);
class UNotification_AchievementUnlocked_TA* SetID(int32_t InID);
};

// Class TAGame.GFxData_Notification_AntiAddiction_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_Notification_AntiAddiction_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Notification_AntiAddiction_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Persona_TA
// 0x0014 (0x0094 - 0x00A8)
class UGFxData_Persona_TA : public UGFxDataRow_X
{
public:
class UPersona_TA* Persona; // 0x0098 (0x0008)
[0x0000000000000000]
class UTexture* PlayerAvatar; // 0x00A0 (0x0008)
[0x000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Persona_TA");
}
}

```

```

return uClassPointer;
};

void HandleGetAvatar(struct FUniqueNetId InPlayerId, class UTexture* Avatar, class FString
OnlinePlayerName);
void UpdatePlayerAvatar();
void RemoveFromDataStore();
void SetPersonaProxy(class UPersona_TA* InPersona);
};

// Class TAGame.GFxData_PartyInviteNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_PartyInviteNotification_TA : public UGFxData_Notification_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PartyInviteNotification_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_PartyMemberProfile_TA
// 0x007C (0x0094 - 0x0110)
class UGFxData_PartyMemberProfile_TA : public UGFxDataRow_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x0098 (0x0048)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FName                   Title;             // 0x00E0 (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString                 TitleText;         // 0x00E8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
struct FColor                  TitleColor;        // 0x00F8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
struct FColor                  TitleGlowColor;   // 0x00FC (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class UOtherPlayerProfile_TA*  Profile;          // 0x0100 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
class UOnlineGameParty_TA*     OnlineGameParty;    // 0x0108
(0x0008) [0x0000800000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PartyMemberProfile_TA");
}

return uClassPointer;
};

bool __GFxData_PartyMemberProfile_TA_SetProfileProducts_0x1(class
UPartyMemberLoadout_TA* P);
void SetProfileProducts();
void HandleLoadoutChanged(struct FUniqueNetId InPlayerId);
void eventOnRemoved();
void Init(struct FUniqueNetId InPlayerId);
};

// Class TAGame.GFxData_PlatformAuth_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_PlatformAuth_TA : public UGFxDataSingleton_X
{
public:
class FString SuccessTitle; // 0x0098 (0x0010)
[0x0000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString SuccessBody; // 0x00A8 (0x0010)
[0x0000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlatformAuth_TA");
}

return uClassPointer;
};

void __GFxData_PlatformAuth_TA_RequestAccountAuthorization_0x1(class UError* Error);
void __GFxData_PlatformAuth_TA_HandleAuthSuccess_0x1(class UError* X);
void AlertSuccess();
void HandleAuthSuccess(class FString AuthToken);
void RequestAccountAuthorization();
};

// Class TAGame.RPC_PlatformAuth_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_PlatformAuth_TA : public URPC_X
{
public:
class FString NintendoAccountID; // 0x00E8 (0x0010)

```

[0x0000004000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_PlatformAuth_TA");  
}  
  
return uClassPointer;  
};  
  
class UErrorType* eventOverrideErrorType(class UErrorType* ErrorType);  
class URPC_PlatformAuth_TA* SetAuthToken(class FString Token);  
};  
  
// Class TAGame.GFxData_PlayerAvatar_TA  
// 0x000C (0x0094 - 0x00A0)  
class UGFxData_PlayerAvatar_TA : public UGFxDataRow_X  
{  
public:  
class UPlayerAvatar_TA* PlayerAvatar; // 0x0098 (0x0008)  
[0x0000004004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerAvatar_TA");  
}  
  
return uClassPointer;  
};  
  
void EquipColor(int32_t InColor);  
void EquipProduct(struct FProductHashID HashID);  
void SetAvatar(class UPlayerAvatar_TA* InPlayerAvatar);  
class FString GetPlayerName();  
struct FUniqueNetId GetPlayerID();  
static class UGFxData_PlayerAvatar_TA* GetOrCreate(class UGFxShell_X* InShell, class  
UPlayerAvatar_TA* InAvatar);  
};  
  
// Class TAGame.GFxData_PlayerRecord_TA  
// 0x008C (0x0094 - 0x0120)  
class UGFxData_PlayerRecord_TA : public UGFxDataRow_X  
{  
public:
```

```

class FString PlayerIDString; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long bReported : 1; // 0x00A8 (0x0004)
[0x0000000040000000] [0x00000001] (CPF>EditInlineNotify)
class FString SanitizedPlayerName; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString PlayerName; // 0x00C0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class APRI_TA* PRI; // 0x00D0 (0x0008)
[0x000004000002000] (CPF_Transient)
struct FUniqueNetId PlayerID; // 0x00D8 (0x0048)
[0x000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerRecord_TA");
}

return uClassPointer;
};

void PrintDebugInfo(class UDebugDrawer* Drawer);
void ReportSeveralReasons(TArray<uint8_t> Reasons);
void Report(uint8_t Reason);
void HandleUniqueIdChanged(class APRI_X* InPRI);
void HandlePlayerNameSanitized(class FString Original, class FString Sanitized);
void HandlePlayerNameChanged(class APRI_X* InPRI);
void eventOnRemoved();
void SetPRI(class APRI_TA* InPRI);
class UOnlineGame_X* GetOnlineGame();
};

// Class TAGame.GFxData_PlayerReport_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_PlayerReport_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PlayerReport_TA");
}

return uClassPointer;
};

```

```

};

void eventOnShellSet();

// Class TAGame.PlayerReportConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UPlayerReportConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FPlayerReportReason> Reasons; // 0x0078 (0x0010)
[0x0000000040400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerReportConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.OnlineGamePlayerTitles_TA
// 0x0050 (0x00B0 - 0x0100)
class UOnlineGamePlayerTitles_TA : public UOnline_X
{
public:
float PlayerTitleCacheTime; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
TArray<struct FCachedPlayerData> PlayerCache; // 0x00B8
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UPlayerTitleConfig_X* TitleConfig; // 0x00C8 (0x0008)
[0x0000800000000000]
struct FScriptDelegate __EventPlayerTitleUpdated__Delegate; // 0x00D0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventPlayerTitlesChanged__Delegate; // 0x00E8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGamePlayerTitles_TA");
}
}

```

```

return uClassPointer;
};

int32_t GetIndex(struct FUniqueNetId& PlayerID);
void HandleOnlineProductsSynced(class USaveData_TA* SaveData);
void HandleSaveDataLoaded(class USaveGameManager_TA* Manager, class USaveData_TA* SaveData, class UError* Error);
TArray<struct FPlayerTitleData> GetPlayerTitleData(struct FUniqueNetId PlayerID);
struct FPlayerTitleData GetTitleData(struct FName TitleId);
TArray<struct FName> GetPlayerTitles(struct FUniqueNetId PlayerID);
void HandlePlayerTitlesRPC(class URPC_GetPlayerTitles_X* RPC);
void SyncPlayerTitles(struct FUniqueNetId PlayerID, struct FScriptDelegate Callback);
void HandlePlayerPsyNetLogin(class UOnlinePlayer_X* Player);
void HandleTitlesChanged();
void OnInit();
void EventPlayerTitlesChanged(class UOnlineGamePlayerTitles_TA* Titles);
void EventPlayerTitleUpdated(class UOnlineGamePlayerTitles_TA* Titles, struct FUniqueNetId PlayerID);
};

// Class TAGame.GFxData_PreMatchLobby_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_PreMatchLobby_TA : public UGFxDataSingleton_X
{
public:
class UPreMatchLobby_TA*           PreMatchLobby;           // 0x0098
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PreMatchLobby_TA");
}

return uClassPointer;
};

void Highlight_Horizontal(int32_t Direction);
void Highlight_Vertical(int32_t Direction);
void ToggleFocus();
void eventOnShellSet();
};

// Class TAGame.GFxData_PremiumGarage_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_PremiumGarage_TA : public UGFxDataSingleton_X
{
public:

public:

```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PremiumGarage_TA");
}

return uClassPointer;
};

class UProfile_TA* GetProfile();
};

// Class TAGame.GFxData_SpectatorStat_TA
// 0x0024 (0x0094 - 0x00B8)
class UGFxData_SpectatorStat_TA : public UGFxDataRow_X
{
public:
class FString           PlayerIDString;          // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           Values;                  // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SpectatorStat_TA");
}

return uClassPointer;
};

void SetPlayerIDString(class FString PlayerID);
};

// Class TAGame.GFxData_ScoreboardStat_TA
// 0x0024 (0x0094 - 0x00B8)
class UGFxData_ScoreboardStat_TA : public UGFxDataRow_X
{
public:
class FString           PlayerIDString;          // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           Values;                  // 0x00A8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
```

```
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ScoreboardStat_TA");
}

return uClassPointer;
};

void SetPlayerIDString(class FString PlayerID);
};

// Class TAGame.GFxData_SpectatorStatNames_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_SpectatorStatNames_TA : public UGFxDataSingleton_X
{
public:
class FString StatNames; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SpectatorStatNames_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_PrivacyPolicy_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_PrivacyPolicy_TA : public UGFxDataSingleton_X
{
public:
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_PrivacyPolicy_TA");
}

return uClassPointer;
};
```

```

bool RequiresAcceptance();
void Accepted();
};

// Class TAGame.PrivateMatchSettingsSave_TA
// 0x0010 (0x0160 - 0x0170)
class UPrivateMatchSettingsSave_TA : public UCustomMatchSettingsSave_TA
{
public:
class FString Region; // 0x0160 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PrivateMatchSettingsSave_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_Privileges_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_Privileges_TA : public UGFxDataSingleton_X
{
public:
unsigned long bCheckingPrivileges : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
class UOnlineGame_X* OnlineGame; // 0x00A0 (0x0008)
[0x0000800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Privileges_TA");
}

return uClassPointer;
};

void HandlePrivilegeCheck(class UPrivilegeCheck_X* PrivilegeCheck);
void PrivilegeCheckFail();
void PrivilegeCheckSuccess();

```

```

void TryToPlayOnlineAndUseUGC();
void TryToUseUGC();
void TryToUsePsyNet();
void TryToPlayOnline();
void TryToViewLeaderboards();
void TryToBrowseInternet();
void UpdateCheckingPrivileges(class UOnlineGamePrivileges_X* Privileges);
void eventOnShellSet();
};

// Class TAGame.GFxData_ProductAttribute_Anthem_TA
// 0x0020 (0x00B0 - 0x00D0)
class UGFxData_ProductAttribute_Anthem_TA : public UGFxData_ProductAttribute_TA
{
public:
class FString           MusicTitle;          // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString           ArtistTitle;         // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Anthem_TA");
}

return uClassPointer;
};

void UpdateAdditionalLoadedData(class UGFxData_Products_TA* GFxData_Products, class ULoadingProduct_TA* ProductData, class UProduct_TA* Product);
};

// Class TAGame.GFxData_ProductAttribute_Blueprint_TA
// 0x001C (0x00B0 - 0x00CC)
class UGFxData_ProductAttribute_Blueprint_TA : public UGFxData_ProductAttribute_TA
{
public:
class UTexture*          ProductThumbnail;    // 0x00B0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class UTexture*          BlueprintThumbnail;   // 0x00B8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t                  Cost;                // 0x00C0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long             bRevealed : 1;       // 0x00C4 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
struct FProductHashID    PreviewProductHashID; // 0x00C8
(0x0004) [0x0000000040000000] (CPF_EditInlineNotify)

public:

```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Blueprint_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Blueprint_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Certified_TA
// 0x0010 (0x00B0 - 0x00C0)
class UGFxData_ProductAttribute_Certified_TA : public UGFxData_ProductAttribute_TA
{
public:
class FString CertifiedStatusLabel; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Certified_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Certified_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Container_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_Container_TA : public UGFxData_ProductAttribute_TA
{
public:
unsigned long bUnlocked : 1; // 0x00B0 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bSkipRoulette : 1; // 0x00B0 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bGiftbox : 1; // 0x00B0 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Container_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Container_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_ContainerKey_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_ContainerKey_TA : public UGFxData_ProductAttribute_TA
{
public:
unsigned long           bMagicKey : 1;           // 0x00B0 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ProductAttribute_ContainerKey_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Key_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_CreatorDetails_TA
// 0x0010 (0x00B0 - 0x00C0)
class UGFxData_ProductAttribute_CreatorDetails_TA : public UGFxData_ProductAttribute_TA
{
public:
class FString           CreatorName;           // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class

```

```

TAGame.GFxData_ProductAttribute_CreatorDetails_TA");
}

return uClassPointer;
};

void UpdateAdditionalLoadedData(class UGFxData_Products_TA* GFxData_Products, class
ULoadingProduct_TA* ProductData, class UProduct_TA* Product);
};

// Class TAGame.ProductAttribute_CreatorDetails_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_CreatorDetails_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_CreatorDetails_TA");
}

return uClassPointer;
};

bool OnInit(class FString AttributeValue, TArray<struct FOnlineProductAttribute>& Attributes);

// Class TAGame.GFxData_ProductAttribute_Currency_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_Currency_TA : public UGFxData_ProductAttribute_TA
{
public:
int32_t          CurrencyID;           // 0x00B0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Currency_TA");
}

return uClassPointer;
};

void UpdateAdditionalLoadedData(class UGFxData_Products_TA* GFxProducts, class

```

```
ULoadingProduct_TA* ProductData, class UProduct_TA* Product);
void OnReceivedAttribute(class UProductAttribute_Currency_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Expiration_TA
// 0x0008 (0x00B0 - 0x00B8)
class UGFxData_ProductAttribute_Expiration_TA : public UGFxData_ProductAttribute_TA
{
public:
    uint64_t           ExpirationTime;          // 0x00B0 (0x0008)
    [0x0000000040000000] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Expiration_TA");
        }

        return uClassPointer;
    }

    void OnReceivedAttribute(class UProductAttribute_Expiration_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Indestructible_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_Indestructible_TA : public UGFxData_ProductAttribute_TA
{
public:
    unsigned long       bIndestructible : 1;      // 0x00B0 (0x0004)
    [0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ProductAttribute_Indestructible_TA");
        }

        return uClassPointer;
    }

    void OnReceivedAttribute(class UProductAttribute_Indestructible_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Level_TA
```

```
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_Level_TA : public UGFxData_ProductAttribute_TA
{
public:
unsigned long          bProductUpgradable : 1;           // 0x00B0 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Level_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Level_TA* Attribute);
};

// Class TAGame.ProductAttribute_LevelSettings_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_LevelSettings_TA : public UProductAttribute_TA
{
public:
TArray<struct FProductLevelData>      Levels;           // 0x0080 (0x0010)
[0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_LevelSettings_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ProductAttribute_Painted_TA
// 0x002C (0x00B0 - 0x00DC)
class UGFxData_ProductAttribute_Painted_TA : public UGFxData_ProductAttribute_TA
{
public:
struct FColor          ProductPaintColor;           // 0x00B0 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
class FString          ProductPaintColorName;        // 0x00B8 (0x0010)
```

```
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture2D*           ProductPaintMask;          // 0x00C8 (0x0008)
[0x0000000040000000] (CPF>EditInlineNotify)
float                      ProductPaintDiffHue;        // 0x00D0 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
float                      ProductPaintDiffSaturation; // 0x00D4 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
float                      ProductPaintDiffLightness; // 0x00D8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Painted_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Painted_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_PostMatchCelebration_TA
// 0x0000 (0x00B0 - 0x00B0)
class UGFxData_ProductAttribute_PostMatchCelebration_TA : public
UGFxData_ProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ProductAttribute_PostMatchCelebration_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ProductAttribute_ProductLink_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_ProductLink_TA : public UGFxData_ProductAttribute_TA
{
public:
```

```

int32_t ProductID; // 0x00B0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ProductAttribute_ProductLink_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_ProductLink_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Reactive_TA
// 0x0030 (0x00B0 - 0x00E0)
class UGFxData_ProductAttribute_Reactive_TA : public UGFxData_ProductAttribute_TA
{
public:
class FString ReactiveLabelText; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString ReactiveDescriptionText; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString ReactiveBulletedDescriptionText; // 0x00D0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Reactive_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Reactive_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Rental_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_Rental_TA : public UGFxData_ProductAttribute_TA
{
public:
int32_t RentalLength; // 0x00B0 (0x0004)

```

[0x0000000040000000] (CPF\_EditInlineNotify)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Rental_TA");  
}  
  
return uClassPointer;  
};  
  
void OnReceivedAttribute(class UProductAttribute_Rental_TA* Attribute);  
};  
  
// Class TAGame.GFxData_ProductAttribute_Schematic_TA  
// 0x0000 (0x00B0 - 0x00B0)  
class UGFxData_ProductAttribute_Schematic_TA : public UGFxData_ProductAttribute_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Schematic_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.GFxData_ProductAttribute_ShopItem_TA  
// 0x0010 (0x00B0 - 0x00C0)  
class UGFxData_ProductAttribute_ShopItem_TA : public UGFxData_ProductAttribute_TA  
{  
public:  
class UTexture* BackgroundImage; // 0x00B0 (0x0008)  
[0x0000000040000000] (CPF_EditInlineNotify)  
class UTexture* ItemLogo; // 0x00B8 (0x0008)  
[0x0000000040000000] (CPF_EditInlineNotify)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_ShopItem_TA");
}

return uClassPointer;
};

void UpdateAdditionalLoadedData(class UGFxData_Products_TA* GFxData_Products, class
ULoadingProduct_TA* ProductData, class UProduct_TA* Product);
};

// Class TAGame.ProductAsset_ShopItem_TA
// 0x0010 (0x00F8100 - 0x01108)
class UProductAsset_ShopItem_TA : public UProductAsset_TA
{
public:
class UTexture*           BackgroundImage;          // 0x00F8100 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UTexture*           ItemLogo;                // 0x01108 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_ShopItem_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_ShopItem_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_ShopItem_TA : public UProductAttribute_TA
{
public:
class UProductAssetReference_TA*      ShopItemReference;    // 0x0080
(0x0008) [0x000000004000001] (CPF_Edit | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_ShopItem_TA");
}
}

```

```

return uClassPointer;
};

bool HasLoadedAllAdditionalReferences(class ULoadingProduct_TA* ProductData);
void LoadAdditionalAssets(struct FScriptDelegate Callback);
bool ShouldLoadAdditionalAssets();
};

// Class TAGame.GFxData_ProductAttribute_SpecialEdition_TA
// 0x0010 (0x00B0 - 0x00C0)
class UGFxData_ProductAttribute_SpecialEdition_TA : public UGFxData_ProductAttribute_TA
{
public:
class FString EditionLabel; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ProductAttribute_SpecialEdition_TA");
}

return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_SpecialEdition_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_TeamEdition_TA
// 0x0028 (0x00B0 - 0x00D8)
class UGFxData_ProductAttribute_TeamEdition_TA : public UGFxData_ProductAttribute_TA
{
public:
class FString TeamName; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture* TeamLogo; // 0x00C0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString ItemShopLabel; // 0x00C8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class

```

```
TAGame.GFxData_ProductAttribute_TeamEdition_TA");
}

return uClassPointer;
};

class UProduct_TA* GetRequiredProduct(class UProduct_TA* Product);
void UpdateAdditionalLoadedData(class UGFxData_Products_TA* GFxData_Products, class
ULoadingProduct_TA* ProductData, class UProduct_TA* Product);
};

// Class TAGame.ProductAsset_ESportsTeam_TA
// 0x0010 (0x00F8100 - 0x01108)
class UProductAsset_ESportsTeam_TA : public UProductAsset_TA
{
public:
    struct FName           TeamName;          // 0x00F8100 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UTexture*        TeamLogo;         // 0x01108 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_ESportsTeam_TA");
        }
    }

    return uClassPointer;
};

// Class TAGame.GFxData_ProductAttribute_Tiered_TA
// 0x0004 (0x00B0 - 0x00B4)
class UGFxData_ProductAttribute_Tiered_TA : public UGFxData_ProductAttribute_TA
{
public:
    int32_t                ProductTier;       // 0x00B0 (0x0004)
    [0x000000040000000] (CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Tiered_TA");
        }
    }
}
```

```
return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_Tiered_TA* Attribute);
};

// Class TAGame.GFxData_ProductAttribute_Title_TA
// 0x0008 (0x00B0 - 0x00B8)
class UGFxData_ProductAttribute_Title_TA : public UGFxData_ProductAttribute_TA
{
public:
    struct FName TitleId; // 0x00B0 (0x0008)
    [0x0000000400000001] (CPF_Edit | CPF_EditInlineNotify)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductAttribute_Title_TA");
        }
    }

    return uClassPointer;
};

void OnReceivedAttribute(class UProductAttribute_TitleID_TA* Attribute);
};

// Class TAGame.GFxData_ProductExpirationNotification_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ProductExpirationNotification_TA : public UGFxData_Notification_TA
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.GFxData_ProductExpirationNotification_TA");
        }
    }

    return uClassPointer;
};

};

// Class TAGame.GFxData_ProductLevelUpgrade_TA
// 0x0000 (0x0098 - 0x0098)
```

```

class UGFxData_ProductLevelUpgrade_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductLevelUpgrade_TA");
}

return uClassPointer;
};

void InitProductData(struct FProductHashID HashID);
};

// Class TAGame.GFxData_ProductsSave_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_ProductsSave_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ProductsSave_TA");
}

return uClassPointer;
};

void ClearUnseenGiftboxes();
void eventOnShellSet();
};

// Class TAGame.RPC_ProductsTradeIn_TA
// 0x0078 (0x00E8 - 0x0160)
class URPC_ProductsTradeIn_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceID>   ProductInstances;    // 0x0130
(0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData>   Drops;           // 0x0140 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

```

```

TArray<class UOnlineProduct_TA*> Products; // 0x0150 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsTradeIn_TA");
}

return uClassPointer;
};

void eventOnComplete();
class URPC_ProductsTradeIn_TA* SetProducts(TArray<struct FProductInstanceId>& InProducts);
class URPC_ProductsTradeIn_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.GFxData_QuickChatBindings_TA
// 0x0058 (0x0098 - 0x00F0)
class UGFxData_QuickChatBindings_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FLocalizedQuickChat> AllQuickChats; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FLocalizedQuickChatBinding> Bindings; // 0x00A8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FName> QuickChats; // 0x00B8 (0x0010)
[0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)
TArray<struct FName> PostGameQuickChats; // 0x00C8
(0x0010) [0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)
TArray<struct FName> PreGameQuickChats; // 0x00D8 (0x0010)
[0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)
class UProfileQuickChatSave_TA* QuickChatSave; // 0x00E8
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_QuickChatBindings_TA");
}

return uClassPointer;
};

struct FLocalizedQuickChat __GFxData_QuickChatBindings_TA__OnShellSet_0x3(struct FName QID);

```

```

struct FLocalizedQuickChat __GFxData_QuickChatBindings_TA__OnShellSet_0x2(struct FName
QID);
struct FLocalizedQuickChat __GFxData_QuickChatBindings_TA__OnShellSet_0x1(struct FName
QID);
int32_t SortQuickChat(struct FName Value1, struct FName Value2);
void GenerateBindings();
void OnBindingsReset();
void ResetBindings();
void ChangeBinding(int32_t Index, struct FName Message);
void HandleSaveLoaded(class UProfileQuickChatSave_TA* InSave);
void OnShellSet();
};

// Class TAGame.GFxData_RedeemCodes_TA
// 0x0028 (0x0098 - 0x00C0)
class UGFxData_RedeemCodes_TA : public UGFxDataSingleton_X
{
public:
class UAsyncTask* RedeemTask; // 0x0098 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString RedeemingTitleString; // 0x00A0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString RedeemingBodyString; // 0x00B0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RedeemCodes_TA");
}

return uClassPointer;
};

void __GFxData_RedeemCodes_TA__Redeem_0x3(class UError* _);
void __GFxData_RedeemCodes_TA__Redeem_0x2(class URPC_RedemCode_TA* RPC);
void __GFxData_RedeemCodes_TA__Redeem_0x1(class URPC_RedemCode_TA* RPC);
void Redeem(class FString InCode);
};

// Class TAGame.RPC_RedemCode_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_RedemCode_TA : public URPC_X
{
public:
class FString Code; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FUniqueNetId PlayerID; // 0x00F8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData> Drops; // 0x0140 (0x0010)

```

[0x0000004000402000] (CPF\_Transient | CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_RedemCode_TA");  
}  
  
return uClassPointer;  
};  
  
class UErrorType* eventOverrideErrorHandler(class UErrorType* ErrorType);  
class URPC_RedemCode_TA* SetPlayerID(struct FUniqueNetId InPlayerId);  
class URPC_RedemCode_TA* SetCode(class FString InCode);  
};  
  
// Class TAGame.GFxData_RegionRestrictions_TA  
// 0x0000 (0x0098 - 0x0098)  
class UGFxData_RegionRestrictions_TA : public UGFxDataSingleton_X  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RegionRestrictions_TA");  
}  
  
return uClassPointer;  
};  
  
bool IsRestricted(uint8_t Restriction);  
};  
  
// Class TAGame.GFxData_Replay_TA  
// 0x0014 (0x0094 - 0x00A8)  
class UGFxData_Replay_TA : public UGFxDataRow_X  
{  
public:  
class FString FriendlyMapName; // 0x0098 (0x0010)  
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Replay_TA");
}

return uClassPointer;
};

void SetReplay(class UReplay_TA* InReplay);
};

// Class TAGame.GFxData_ReplayViewer_TA
// 0x0088 (0x0098 - 0x0120)
class UGFxData_ReplayViewer_TA : public UGFxDataSingleton_X
{
public:
int32_t CurrentFrame; // 0x0098 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify) CameraModes; // 0x00A0
TArray<struct FLocalizedCameraMode> (CPF_NeedCtorLink | CPF_EditInlineNotify)
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
struct FName DefaultCameraMode; // 0x00B0 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
struct FName CameraMode; // 0x00B8 (0x0008)
[0x0000000040000000] (CPF_EditInlineNotify)
class FString FocusActorString; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString LastFocusActorString; // 0x00D0 (0x0010)
[0x00000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long bHighlightFocusActor : 1; // 0x00E0 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long bShowReplayHUD : 1; // 0x00E0 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)
unsigned long bShowMatchInfoHUD : 1; // 0x00E0 (0x0004)
[0x0000000040000000] [0x00000004] (CPF_EditInlineNotify)
unsigned long bShowPlayerNames : 1; // 0x00E0 (0x0004)
[0x0000000040000000] [0x00000008] (CPF_EditInlineNotify)
unsigned long bPausedForScrub : 1; // 0x00E0 (0x0004)
[0x00000000000000002000] [0x00000010] (CPF_Transient)
unsigned long bChangesMade : 1; // 0x00E0 (0x0004)
[0x0000000040000000] [0x00000020] (CPF_EditInlineNotify)
float Slomo; // 0x00E4 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
float FOV; // 0x00E8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t ViewRoll; // 0x00EC (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UReplay_TA* Replay; // 0x00F0 (0x0008)
[0x00000000000000002000] (CPF_Transient)
int32_t PendingSkipToFrame; // 0x00F8 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UGFxModal_X* ModalProcessing; // 0x0100 (0x0008)
[0x00000000000000002000] (CPF_Transient)
struct FScriptDelegate __EventHudVisibilityUpdated__Delegate; // 0x0108

```

(0x0018) [0x0000000000400000] (CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ReplayViewer_TA");  
}  
  
return uClassPointer;  
};  
  
void AddCameraTrackPoint();  
void HandleCameraTargetChanged(class ACamera_TA* InCamera, class AActor* InTarget);  
void HandleReplayExported(class UReplayManager_TA* Manager, class FString Id, class UError* Error);  
void SaveReplayAndExit();  
void SetPausedForScrub(unsigned long bPaused);  
void OnTimelineScrub();  
void OpenReplayFXMenu();  
void OpenCameraMenu(int32_t KeyCode, int32_t MouseButton);  
void OpenFocusMenu(int32_t KeyCode, int32_t MouseButton);  
void OpenSpeedMenu(int32_t KeyCode, int32_t MouseButton);  
void RemoveKeyframe(int32_t KeyframeIndex);  
void AddKeyframe(struct FName Type);  
void SetFOV(float InFOV);  
void SetShowPlayerNames(unsigned long bVisible);  
void SetShowMatchInfoHUD(unsigned long bVisible);  
void SetShowReplayHUD(unsigned long bVisible);  
void SetSlomo(float NewSlomo);  
void TimerSkipToFrame();  
void AdvanceTime(float DeltaTime);  
void AdvanceToFrame(int32_t frame);  
void SetCameraMode(struct FName Mode);  
void SetFocusActorString(class FString InFocusActorString);  
void RefreshDisabledModes();  
void InitCameraModes();  
void UpdateReplayData();  
void SetReplay(class UReplay_TA* InReplay);  
void EventHudVisibilityUpdated();  
};  
  
// Class TAGame.GFxData_Restrictions_TA  
// 0x0010 (0x0098 - 0x00A8)  
class UGFxData_Restrictions_TA : public UGFxDataSingleton_X  
{  
public:  
class UGFxModal_X* [0x0000004000000000] RestrictedModal; // 0x0098 (0x0008)  
class UEpicConfig_X* [0x0000804000000000] EpicConfig; // 0x00A0 (0x0008)
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Restrictions_TA");
}

return uClassPointer;
};

void OnOk(class UGFxModal_X* Modal);
void OnPrivacyPolicy(class UGFxModal_X* Modal);
bool IsRestrictedNoModal(uint8_t Reason);
bool IsRestricted(uint8_t Reason);
bool IsRestrictedInternal(uint8_t Reason);
};

// Class TAGame.UIUtil_TA
// 0x0000 (0x0098 - 0x0098)
class UUIUtil_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UIUtil_TA");
}

return uClassPointer;
};

static class FString GetClickableUrl(class FString URL, class FString Text);
};

// Class TAGame.GFxModal_Info_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Info_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Info_TA");
}

return uClassPointer;
};

class UGFxModal_Info_TA* SetInfoText(class FString Value);
class UGFxModal_Info_TA* SetHeaderVisible(unsigned long Value);
void HandleButtonClicked(int32_t Index);
};

// Class TAGame.GFxData_RewardDrop_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_RewardDrop_TA : public UGFxDataSingleton_X
{
public:
class UObjectProvider* RewardDrops; // 0x0098 (0x0008)
[0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RewardDrop_TA");
}

return uClassPointer;
};

void __GFxData_RewardDrop_TA__OnShellSet_0x1(class UObject* _);
void OnAllDropsHandled();
void OnReceivedDrop();
void ClearDrops();
void GetNextDrop();
bool HasDrops();
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_RewardDrop_XP_TA
// 0x0004 (0x00A0 - 0x00A4)
class UGFxData_RewardDrop_XP_TA : public UGFxData_RewardDrop_TA
{
public:
unsigned long bCanDropOnlineRewards : 1; // 0x00A0 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RewardDrop_XP_TA");
}

return uClassPointer;
};

void __GFxData_RewardDrop_XP_TA__OnShellSet_0x1(class AGameEvent_Soccar_TA*
SoccarGame);
void __GFxData_RewardDrop_XP_TA__OnShellSet_0x2();
void SetTotalGainedRTPC(float Value);
void SetBarPercentageRTPC(float Value);
void eventOnShellSet();
};

// Class TAGame.GFxData_RewardDropGroup_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_RewardDropGroup_TA : public UGFxData_ItemDropGroup_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RewardDropGroup_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_RewardDrops_TA
// 0x00AC (0x0098 - 0x0144)
class UGFxData_RewardDrops_TA : public UGFxDataSingleton_X
{
public:
struct FName RewardName; // 0x0098 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t Total; // 0x00A0 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t Base; // 0x00A4 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t BaseLevel; // 0x00A8 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t Level; // 0x00AC (0x0004)
}

```

```

[0x0001000040000000] (CPF_EditInlineNotify)
TArray<struct FGFxLevelThreshold>           LevelThresholds;          // 0x00B0
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FOnlineXPMODIFIER>           Modifiers;              // 0x00C0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UObjectProvider*                      RewardDrops;            // 0x00D0 (0x0008)
[0x0001000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF>EditInline)
TArray<struct FName>                      PremiumModifierNames;    // 0x00D8
(0x0010) [0x0001000000400001] (CPF>Edit | CPF_NeedCtorLink)
TArray<struct FName>                      PenaltyModifierNames;   // 0x00E8
(0x0010) [0x0001000000400001] (CPF>Edit | CPF_NeedCtorLink)
TArray<struct FName>                      PromoModifierNames;     // 0x00F8
(0x0010) [0x0001000000400003] (CPF>Edit | CPF_Const | CPF_NeedCtorLink)
struct FName                               WeeklyCapReached;        // 0x0108 (0x0008)
[0x0001000000000002] (CPF_Const)
class FString                            RewardModifiers_SectionName; // 0x0110 (0x0010)
[0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString                            RewardModifiers_PackageName; // 0x0120
(0x0010) [0x0001000000400002] (CPF_Const | CPF_NeedCtorLink)
struct FName                               WeeklyCap;               // 0x0130 (0x0008)
[0x0001000000000002] (CPF_Const)
struct FName                               RestXP;                 // 0x0138 (0x0008)
[0x0001000000000002] (CPF_Const)
unsigned long                            bCanDropOnlineRewards : 1; // 0x0140 (0x0004)
[0x0001000000000000] [0x00000001]

```

public:

```

static UClass* StaticClass()
{
    static UClass* uClassPointer = nullptr;

```

```
if (!uClassPointer)
```

```
{
    uClassPointer = UObject::FindClass("Class TAGame.GFxData_RewardDrops_TA");
}
```

```
return uClassPointer;
};
```

```
void __GFxDATA_RewardDrops_TA__OnShellSet_0x2(class AGameEvent_Soccar_TA*
SoccarGame);
```

```
void __GFxDATA_RewardDrops_TA__OnShellSet_0x3();
```

```
void __GFxDATA_RewardDrops_TA__OnShellSet_0x1(class UObject* _);
```

```
bool ShouldShowRewards();
```

```
void SetBarPercentageRTPC(float Value);
```

```
void SetTotalGainedRTPC(float Value);
```

```
void OnAllRewardsHandled();
```

```
void SetAllRewardsHandled();
```

```
void GetNextReward();
```

```
void OnReceivedReward();
```

```
bool HasRewards();
```

```
class FString LocalizeModifier(struct FOnlineXPMODIFIER Modifier);
```

```
void UpdateXPReward(class URewardDrop_XP_TA* XP);
```

```
void UpdateRewardData();
```

```

void OnRewardAddedToQueue();
void eventOnShellSet();
};

// Class TAGame.GFxData_RocketPass_TA
// 0x00A8 (0x0098 - 0x0140)
class UGFxData_RocketPass_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FRocketPassProduct>           FreeRewards;          // 0x0098
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FRocketPassProduct>           PremiumRewards;      // 0x00A8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FRocketPassProduct>           PrestigeRewards;     // 0x00B8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FRocketPassXPReward>          PremiumXPRewards;    // 0x00C8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FRocketPassCurrencyReward>      PremiumCurrencyRewards; // 0x00D8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint64_t                         StartTime;           // 0x00E8 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
uint64_t                         EndTime;            // 0x00F0 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
class UOnlineProductStoreSet_TA*        RocketPassSet;       // 0x00F8
(0x0008) [0x0001000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UAsyncTask*                     PendingMasterTask;   // 0x0100 (0x0008)
[0x0001000000002000] (CPF_Transient)
class FString                        RegionRestrictedPurchaseTiers; // 0x0108 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
struct FName                          MainMenuName;       // 0x0118 (0x0008)
[0x0001000000000002] (CPF_Const)
struct FName                          RocketPassMenuName; // 0x0120 (0x0008)
[0x0001000000000002] (CPF_Const)
unsigned long                        bHideMainMenuButton : 1; // 0x0128 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
class URocketPass_TA*                RocketPass;         // 0x0130 (0x0008)
[0x0001000000002001] (CPF>Edit | CPF_Transient)
class URocketPassConfig_TA*         RocketPassConfig;   // 0x0138
(0x0008) [0x0001800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RocketPass_TA");
}

return uClassPointer;
};

```

```

void __GFxData_RocketPass_TA_HandleRocketPassChange_0x3();
void __GFxData_RocketPass_TA_HandleRocketPassChange_0x2();
void __GFxData_RocketPass_TA_HandleRocketPassChange_0x1();
void __GFxData_RocketPass_TA_GetRocketPassData_0x1();
void HandleError(class UError* InError);
void OnCanOpenMenu();
void GetRocketPassData();
void HandlePrivilegeCheck(class UPrivilegeCheck_X* PrivilegeCheck);
void OpenMenu();
void RequestPrestigeRewards();
void UpdateRewards(TArray<struct FRocketPassRewardData>& RewardData, TArray<struct FRocketPassProduct>& OutProducts, TArray<struct FRocketPassXPReward>& OutXPRewards, TArray<struct FRocketPassCurrencyReward>& OutCurrencyRewards);
void PurchaseTiers(TArray<struct FProductInstanceID> KeyInstanceIDs, int32_t PurchasableID);
void PurchasePremium(TArray<struct FProductInstanceID> KeyInstanceIDs, int32_t PurchasableID);
void HandleRocketPassConfigChange();
void HandleRocketPassInfoChanged(class URocketPass_TA* InRocketPass);
void HandleTopMenuChanged(struct FName PrevMenu, struct FName TopMenu, unsigned long bClearingStack);
void HandleRocketPassChange(class URocketPass_TA* InRocketPass);
void eventOnRemoved();
void eventOnShellSet();
};


```

```

// Class TAGame.RocketPassConfig_TA
// 0x0048 (0x0078 - 0x00C0)
class URocketPassConfig_TA : public UOnlineConfig_X
{
public:
int32_t RocketPassID; // 0x0078 (0x0004)
[0x0001000000000000]
uint64_t StartTime; // 0x0080 (0x0008)
[0x0001000000000000]
uint64_t EndTime; // 0x0088 (0x0008)
[0x0001000000000000]
int32_t StartRangeSeconds; // 0x0090 (0x0004)
[0x0001000000000000]
uint64_t FinalStartTime; // 0x0098 (0x0008)
[0x0001000000000000]
unsigned long bIsActive : 1; // 0x00A0 (0x0004)
[0x0009000000002000] [0x00000001] (CPF_Transient)
unsigned long bHideMainMenuButton : 1; // 0x00A0 (0x0004)
[0x0001000000000000] [0x00000002]
struct FScriptDelegate __bIsActive__ChangeNotify; // 0x00A8
(0x0018) [0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.RocketPassConfig_TA");
}

return uClassPointer;
};

void __bIsActive__ChangeNotifyFunc();
void UpdateActiveState();
bool IsActive();
uint64_t GetFinalStartTime();
bool IsValid();
void Apply();
};

// Class TAGame.GFxData_RocketPassBundleContainer_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_RocketPassBundleContainer_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_RocketPassBundleContainer_TA");
}

return uClassPointer;
};

void eventOnShellSet();
};

// Class TAGame.RocketPassBundleContainer_TA
// 0x0030 (0x0070 - 0x00A0)
class URocketPassBundleContainer_TA : public UComponent
{
public:
struct FName RocketPassMenuName; // 0x0070 (0x0008)
[0x0001000000000003] (CPF_Edit | CPF_Const)
struct FName ChallengeMenuName; // 0x0078 (0x0008)
[0x0001000000000003] (CPF_Edit | CPF_Const)
TArray<struct FRocketPassBundleInfo> Premium; // 0x0080
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FRocketPassBundleInfo> Tier; // 0x0090 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RocketPassBundleContainer_TA");
}

return uClassPointer;
};

class UGFxData_MenuStack_TA* GetMenuStack();
void HandleImageDownloaded(struct FOnlineImageDownload DownloadResult);
void UpdateBundleImages(class UOnlineImageDownloaderWeb* ImageDownloader,
TArray<struct FRocketPassBundleInfo>& Bundles, TArray<class FString>& RequiredURLs);
void DownloadBundleImages();
void HandleTopMenuChanged(struct FName PrevMenu, struct FName TopMenu, unsigned long bClearingStack);
void SetBundleInfo(struct FRocketPassStore RPInfo, class URocketPass_TA* InRocketPass);
};

// Class TAGame.GFxData_SchematicTradeIn_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_SchematicTradeIn_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FSchematicResourceData> Schematics; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UGFxModal_X* ModalProcessing; // 0x00A8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString InvalidProductString; // 0x00B0 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
int32_t MinResources; // 0x00C0 (0x0004)
[0x0000000000000000]
int32_t MaxResources; // 0x00C4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SchematicTradeIn_TA");
}

return uClassPointer;
};

class UProfile_TA* GetProfile();
int32_t GetRemainingResources();
bool IsProcessingTrade();
void OnSchematicTradeInFailed();
void OnSchematicTradeInComplete();
void HandleTradeInComplete(class URPC_SchematicsTradeIn_TA* RPC);

```

```

void SetSchematicResource(struct FProductInstanceID ProductInstance, int32_t Amount);
void IncrementSchematicResource(struct FProductInstanceID ProductInstance, int32_t Amount);
void RemoveSchematic(struct FProductInstanceID ProductInstance);
bool AddSchematic(struct FProductInstanceID ProductInstance);
void TradeIn(struct FProductInstanceID InContainerInstanceID);
};

// Class TAGame.RPC_SchematicsTradeIn_TA
// 0x00A8 (0x00E8 - 0x0190)
class URPC_SchematicsTradeIn_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;                      // 0x00E8 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)
    struct FProductInstanceID      CrateID;                      // 0x0130 (0x0010)
    [0x0000004000000000]
    TArray<struct FSchematicResourceData>   Schematics;        // 0x0140
    (0x0010) [0x0000004000400000] (CPF_NeedCtorLink)
    TArray<struct FOnlineProductData>   Drops;                  // 0x0150 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FOnlineProductData>   CrateDrops;          // 0x0160 (0x0010)
    [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FProductInstanceID>     ConsumedInstanceIDs; // 0x0170
    (0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<class UOnlineProduct_TA*>     Products;            // 0x0180 (0x0010)
    [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_SchematicsTradeIn_TA");
        }

        return uClassPointer;
    };

    void eventOnComplete();
    class URPC_SchematicsTradeIn_TA* SetSchematics(TArray<struct FSchematicResourceData>& InSchematics);
    class URPC_SchematicsTradeIn_TA* SetCrateID(struct FProductInstanceID InCrateID);
    class URPC_SchematicsTradeIn_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.GFxData_Scoreboard_TA
// 0x0028 (0x0098 - 0x00C0)
class UGFxData_Scoreboard_TA : public UGFxDataSingleton_X
{
public:
    unsigned long           bSplitByTeams : 1;                // 0x0098 (0x0004)
    [0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)

```

```

class FString StatNames; // 0x00A0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString SortedPlayerIDs; // 0x00B0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Scoreboard_TA");
}

return uClassPointer;
};

class FString __GFxData_Scoreboard_TA__UpdateSortedPlayerIDs_0x2(class UGFxData_PRI_TA* P);
bool __GFxData_Scoreboard_TA__UpdateSortedPlayerIDs_0x1(class UGFxData_PRI_TA* P);
void UpdateSortedPlayerIDs(TArray<class UGFxData_PRI_TA*>& SortedPRIData);
};

// Class TAGame.GFxData_Season_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_Season_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FSeasonTeamRank> TeamRankings; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FSeasonMatch> WeekMatches; // 0x00A8
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FSeasonWeek> Weeks; // 0x00B8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UProfile_TA* Profile; // 0x00C8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class USeason_TA* Season; // 0x00D0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Season_TA");
}

return uClassPointer;
};

int32_t SortWeekMatch(struct FSeasonMatch A, struct FSeasonMatch B);

```

```
void SetDifficulty(int32_t NewDifficulty);
void SetBots(class FString BotID0, class FString BotName0, class FString BotID1, class FString
BotName1, class FString BotID2, class FString BotName2);
void LoadWeekMatches(int32_t Week);
void CalcCurrentWeekRankings();
void UpdateSeasonWeeks();
void HandleWeekSimulated(class USeason_TA* InSeason);
void SetSeason(class USeason_TA* InSeason);
void RelocalizeTeams();
void LoadLatestSeason();
void eventOnShellSet();
};

// Class TAGame.ProductAttribute_Trademark_TA
// 0x0001 (0x0080 - 0x0081)
class UProductAttribute_Trademark_TA : public UProductAttribute_TA
{
public:
    uint8_t                      Glyph;                                // 0x0080 (0x0001)
    [0x0000000000000003] (CPF_Edit | CPF_Const)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_Trademark_TA");
        }

        return uClassPointer;
    }

    class FString GetGlyph();
};

// Class TAGame.ProductAttribute_DisplayLabelSlot_TA
// 0x0008 (0x0080 - 0x0088)
class UProductAttribute_DisplayLabelSlot_TA : public UProductAttribute_TA
{
public:
    class UProductSlot_TA*          DisplayLabelSlot;                // 0x0080 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_DisplayLabelSlot_TA");
        }

        return uClassPointer;
    }
}
```

```

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_CannotBeArchived_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_CannotBeArchived_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_CannotBeArchived_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_SeasonMode_TA
// 0x0058 (0x0098 - 0x00F0)
class UGFxData_SeasonMode_TA : public UGFxDataSingleton_X
{
public:
TArray<class UProduct_TA*> PlayerBots; // 0x0098 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
int32_t NumSeasons; // 0x00A8 (0x0004)
[0x0000000040000000] (CPF>EditInlineNotify)
TArray<struct FGFxLogoData> Logos; // 0x00B0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FGFxBotData> Bots; // 0x00C0 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UProductAsset_Logo_TA* TeamLogo; // 0x00D0 (0x0008)
[0x0000000000002000] (CPF_Transient)
class USeasonSave_TA* SeasonSave; // 0x00D8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UCarPreviewSet_TA* PreviewSet; // 0x00E0 (0x0008)
[0x0000000000002000] (CPF_Transient)
unsigned long bUseSeasonColors : 1; // 0x00E8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
int32_t CurrentTeamSize; // 0x00EC (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SeasonMode_TA");
}

return uClassPointer;
};

bool InMainMenu();
void DeleteSeason();
void CreateSeason(int32_t TeamSize, int32_t Difficulty, int32_t NumRegularWeeks, int32_t
NumPlayoffsTeams, class FString TeamName, struct FName TeamLogoAsset, class FString
BotID0, class FString BotID1, class FString BotID2);
void ClearBotPreviewActors();
void TeamSizeChanged(int32_t TeamSize);
void SetBot(int32_t BotIndex, class FString BotID);
void UpdateColors(class ACarPreviewActor_TA* PreviewActor);
void RestoreStadiumColors();
void SetLogo(struct FName LogoAsset);
class FString GetBotNameFromID(class FString Id);
void BuildBotData();
void BuildLogoData();
void HandlePreviewActorsChanged(class UCarPreviewSet_TA* G);
void HandleSetProfile(class ACarPreviewActor_TA* Car);
void eventOnShellSet();
};

// Class TAGame.GFxData_SeasonReward_TA
// 0x0020 (0x0098 - 0x00B8)
class UGFxData_SeasonReward_TA : public UGFxDataSingleton_X
{
public:
int32_t Level; // 0x0098 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t Wins; // 0x009C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
int32_t WinsRequired; // 0x00A0 (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long bCanProgress : 1; // 0x00A4 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
class URankedConfig_X* RankedConfig; // 0x00A8 (0x0008)
[0x0000800000000000]
class UOnlineGameSkill_X* Skill; // 0x00B0 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.GFxData_SeasonReward_TA");
}

return uClassPointer;
};

void UpdateData();
void HandleRewardUpdateOnGameJoin(unsigned long bSuccess, class FString FailReason);
void HandleSkillObjectInjected();
};

// Class TAGame.ServerBrowserSettingsSave_TA
// 0x0000 (0x0160 - 0x0160)
class UServerBrowserSettingsSave_TA : public UCustomMatchSettingsSave_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerBrowserSettingsSave_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ServerPerformanceStats_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_ServerPerformanceStats_TA : public UGFxDataSingleton_X
{
public:
uint8_t ServerPerformanceStatus; // 0x0098 (0x0001)
[0x0000000040000000] (CPF_EditInlineNotify)
unsigned long bReportedServer : 1; // 0x009C (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ServerPerformanceStats_TA");
}

return uClassPointer;
};

```

```

void ReportServer();
void UpdateServerPerformanceStatus(uint8_t Status);
};

// Class TAGame.UserSettingObserver_TA
// 0x0010 (0x0070 - 0x0080)
class UUserSettingObserver_TA : public UComponent
{
public:
class UUserSettingObserverEventCache_TA* Cache; // 0x0070
(0x0008) [0x0001000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UUserSettingMetrics_TA* Metrics; // 0x0078 (0x0008)
[0x0001000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserSettingObserver_TA");
}

return uClassPointer;
};

void __UserSettingObserver_TA__HandleCacheTimeout_0x1(struct FSettingEvent E);
void HandleScalarValueSet(class UGFxData_UserSetting_TA* Setting, float Value);
void HandleBooleanValueSet(class UGFxData_UserSetting_TA* Setting, unsigned long Value);
void HandleListValueSet(class UGFxData_UserSetting_TA* Setting, class FString Value);
void HandleCacheTimeout(class UUserSettingObserverEventCache_TA* InCache);
void RecordSetting(class UGFxData_UserSetting_TA* Setting);
void AddListener(class UGFxData_UserSetting_TA* Setting);
void AddListeners(TArray<class UGFxData_UserSetting_TA*>& UserSettings);
void Init();
};

// Class TAGame.UserSettingMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UUserSettingMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserSettingMetrics_TA");
}

```

```

}

return uClassPointer;
};

void Changed(struct FUniqueNetId PlayerID, struct FName Id, class FString Value);

// Class TAGame.VideoSettingsSave_TA
// 0x0014 (0x00C8 - 0x00DC)
class UVideoSettingsSave_TA : public USaveObject_TA
{
public:
    unsigned long          bVsync : 1;           // 0x00C8 (0x0004)
    [0x0000000000000000] [0x00000001]
    unsigned long          bShowWeatherFX : 1;    // 0x00C8 (0x0004)
    [0x0000000000000000] [0x00000002]
    unsigned long          bShowLightShafts : 1;   // 0x00C8 (0x0004)
    [0x0000000000000000] [0x00000004]
    unsigned long          bShowLensFlares : 1;    // 0x00C8 (0x0004)
    [0x0000000000000000] [0x00000008]
    unsigned long          bEnableHDRSideBySideVisualizer : 1; // 0x00C8
    (0x0004) [0x0000000000000000] [0x00000010]
    unsigned long          bUncappedFramerate : 1;    // 0x00C8 (0x0004)
    [0x0000000000000000] [0x00000020]
    float                 HDRBrightnessScale;        // 0x00CC (0x0004)
    [0x0000000000000000]
    float                 HDRPaperWhiteScale;        // 0x00D0 (0x0004)
    [0x0000000000000000]
    float                 HDRGammaScale;            // 0x00D4 (0x0004)
    [0x0000000000000000]
    float                 CustomFPS;                // 0x00D8 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.VideoSettingsSave_TA");
        }
    }

    return uClassPointer;
};

};

// Class TAGame.SoundSettingsControllerSave_TA
// 0x0038 (0x00C8 - 0x0100)
class USoundSettingsControllerSave_TA : public USaveObject_TA
{
public:

```

```
float Volume; // 0x00C8 (0x0004)
[0x0008000000000000]
uint8_t SoundMode; // 0x00CC (0x0001)
[0x0008000000000000]
struct FScriptDelegate __Volume_ChangeNotify; // 0x00D0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __SoundMode_ChangeNotify; // 0x00E8
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SoundSettingsControllerSave_TA");
}

return uClassPointer;
};

void __SoundMode_ChangeNotifyFunc();
void __Volume_ChangeNotifyFunc();
};

// Class TAGame.SettingsMenuConfig_TA
// 0x0004 (0x0078 - 0x007C)
class USettingsMenuConfig_TA : public UOnlineConfig_X
{
public:
unsigned long bShowTrainingTab : 1; // 0x0078 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SettingsMenuConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.VideoSettingsSavePC_TA
// 0x0030 (0x00DC - 0x010C)
class UVideoSettingsSavePC_TA : public UVideoSettingsSave_TA
{
public:
```

```
int32_t WindowMode; // 0x00E0 (0x0004)
[0x0000000000000000]
TArray<struct FOptionsValue> VideoOptions; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString Resolution; // 0x00F8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t MaxFPS; // 0x0108 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VideoSettingsSavePC_TA");
}

return uClassPointer;
};

};

// Class TAGame.VideoSettingsSaveConsole_TA
// 0x0005 (0x00DC - 0x00E1)
class UVideoSettingsSaveConsole_TA : public UVideoSettingsSave_TA
{
public:
uint8_t VideoQualityMode; // 0x00E0 (0x0001)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VideoSettingsSaveConsole_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_WeatherToggled_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_WeatherToggled_TA : public USequenceEvent
{
public:

public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_WeatherToggled_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_UserSettingChanged_TA
// 0x000C (0x017C - 0x0188)
class USeqEvent_UserSettingChanged_TA : public USequenceEvent
{
public:
struct FName           SettingName;          // 0x0180 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_UserSettingChanged_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ShopAds_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_ShopAds_TA : public UGFxDataSingleton_X
{
public:
class UShopAdsConfig_TA*      Config;          // 0x0098 (0x0008)
[0x00010800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ShopAds_TA");
}
}
```

```

return uClassPointer;
};

void HandleConfigChanged();
};

// Class TAGame.ShopsConfig_TA
// 0x0018 (0x0078 - 0x0090)
class UShopsConfig_TA : public UOnlineConfig_X
{
public:
int32_t DebugShopID; // 0x0078 (0x0004)
[0x0001000000000000]
unsigned long bEnableShopMetrics : 1; // 0x007C (0x0004)
[0x0001000000000000] [0x00000001]
unsigned long bHideMainMenuButton : 1; // 0x007C (0x0004)
[0x0001000000000000] [0x00000002]
TArray<class FString> ShopTypesForOwnedTracking; // 0x0080
(0x0010) [0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopsConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.ShopAutoLoadHelper_TA
// 0x0048C (0x0060 - 0x00A8C)
class UShopAutoLoadHelper_TA : public UObject
{
public:
int32_t SyncRequestRange; // 0x0060 (0x0004)
[0x00010004000004000] (CPF_Config)
class URetryDelay_X* AutoLoadDelay; // 0x0068 (0x0008)
[0x00010000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class UASyncTask* AutoLoadAsyncTask; // 0x0070 (0x0008)
[0x00010000000002000] (CPF_Transient)
TArray<struct FShopItem> AutoLoadedItems; // 0x0078 (0x0010)
[0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UShopItemExpirationHelper_TA* AutoLoadExpirationHelper; // 0x0088
(0x0008) [0x00010000000000000]
uint64_t MinEndDateOffset; // 0x0090 (0x0008)
[0x00010000000000000]
uint64_t MaxEndDateOffset; // 0x0098 (0x0008)

```

```
[0x0001000000000000]           RandomEndDateOffset;           // 0x00A0 (0x0008)
uint64_t                         RandomEndDateOffset;
[0x0000000000000000]           LastRequestStartTime;        // 0x00A8 (0x0004)
float                            LastRequestStartTime;
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopAutoLoadHelper_TA");
}

return uClassPointer;
};

void __ShopAutoLoadHelper_TA__Construct_0x2(class UShopItemExpirationHelper_TA* _);
void __ShopAutoLoadHelper_TA__Construct_0x1(class UShopItemExpirationHelper_TA* _);
void __ShopAutoLoadHelper_TA__CheckAutoLoadSucceeded_0x1(class UError* _);
static bool __ShopAutoLoadHelper_TA__HasExpiredItems_0x2(struct FShopItem Item);
static bool __ShopAutoLoadHelper_TA__HasExpiredItems_0x1(struct FShopItem Item);
void OnLoadedChanged();
void OnAutoReloadChanged();
void RefreshAutoLoadedItems();
bool LoadInProgress();
static bool HasExpiredItems(TArray<struct FShopItem>& InItems);
static TArray<struct FShopItem> ApplyEndDateOffset(uint64_t Offset, TArray<struct FShopItem>& InItems);
void HandleShopItemExpired();
void HandleAutoLoadAbandoned();
void HandleAutoLoadFailed(class UError* Error);
void CheckAutoLoadSucceeded(TArray<struct FShopItem> InAutoLoadedItems);
void AutoLoadCatalogue();
TArray<struct FShopItem> GetShopItems();
void eventConstruct();
};

// Class TAGame.ShopItemExpirationHelper_TA
// 0x0028 (0x0060 - 0x0088)
class UShopItemExpirationHelper_TA : public UObject
{
public:
TArray<uint64_t>                 ShopItemExpirations;          // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate             __EventShopItemExpired__Delegate; // 0x0070
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopItemExpirationHelper_TA");
}

return uClassPointer;
};

uint64_t __ShopItemExpirationHelper_TA__UpdateExpirations_0x2(struct FShopItem Item);
bool __ShopItemExpirationHelper_TA__UpdateExpirations_0x1(struct FShopItem Item);
void HandleItemExpired();
void UpdateExpirationTimer();
void UpdateExpirations(TArray<struct FShopItem>& ShopItems);
void ClearShopItems();
void SetShopItems(TArray<struct FShopItem> ShopItems);
void EventShopItemExpired(class UShopItemExpirationHelper_TA* ShopExpirationHelper);
};

// Class TAGame.LoadedMtxCatalog_TA
// 0x0020 (0x0070 - 0x0090)
class ULoadedMtxCatalog_TA : public UComponent
{
public:
TArray<uint8_t> ValidMtxShopTabs; // 0x0070 (0x0010)
[0x0001000400040001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FLoadedMtxCatalog> Catalogs; // 0x0080 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadedMtxCatalog_TA");
}

return uClassPointer;
};

void HandleMtxCatalogLoaded(struct FMTXPurchaseInfo Info, uint8_t Category);
};

// Class TAGame.ShopTabs_TA
// 0x0020 (0x0070 - 0x0090)
class UShopTabs_TA : public UComponent
{
public:
class UShopTabsConfig_TA* Config; // 0x0070 (0x0008)
[0x0001080000000001] (CPF_Edit)
class ULoadedMtxCatalog_TA* LoadedMtx; // 0x0078 (0x0008)
[0x0001000000408009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

```

```

class FString EsportsLabel; // 0x0080 (0x0010)
[0x00010000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopTabs_TA");
}

return uClassPointer;
};

struct FShopTab __ShopTabs_TA__GetAvailableShopTabs_0x2(struct FLoadedMtxCatalog Item);
bool __ShopTabs_TA__GetAvailableShopTabs_0x1(class UGFxData_ShopCatalogue_TA* Shop);
TArray<struct FShopTab> GetAvailableShopTabs(TArray<class UGFxData_ShopCatalogue_TA*> Shops);
void eventConstruct();
};

// Class TAGame.ShopTabsConfig_TA
// 0x0068 (0x0078 - 0x00E0)
class UShopTabsConfig_TA : public UOnlineConfig_X
{
public:
unsigned long bUseLegacyTabSorting : 1; // 0x0078 (0x0004)
[0x0001000000000000] [0x00000001]
unsigned long bDisableItemsInCatalogCheck : 1; // 0x0078 (0x0004)
[0x0001000000000001] [0x00000002] (CPF_Edit)
unsigned long bDebugShowIds : 1; // 0x0078 (0x0004)
[0x0001000000000001] [0x00000004] (CPF_Edit)
unsigned long bDebugVerboseDebugging : 1; // 0x0078 (0x0004)
[0x0001000000000001] [0x00000008] (CPF_Edit)
TArray<class FString> EsportsShopValidRegions; // 0x0080
(0x0010) [0x00010004000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString ShopType_Esports; // 0x0090 (0x0010)
[0x00010004000400000] (CPF_NeedCtorLink)
class FString ShopType_EsportsTeam; // 0x00A0 (0x0010)
[0x00010004000400000] (CPF_NeedCtorLink)
class FString ShopType_MTX; // 0x00B0 (0x0010)
[0x00010004000400000] (CPF_NeedCtorLink)
TArray<class FString> DisabledTypes; // 0x00C0 (0x0010)
[0x000001000000400001] (CPF_Edit | CPF_NeedCtorLink)
class UIEpochNow* Time_Object; // 0x00D0 (0x0008)
[0x0001000000000000]
class UIEpochNow* Time_Interface; // 0x00D8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopTabsConfig_TA");
}

return uClassPointer;
};

class FString BuildIdLabel(int32_t ItemID, class FString DefaultLabel);
bool ValidateShop(class UGFxData_ShopCatalogue_TA* Shop);
bool IsValidEsportsTeamRegion(class FString InRegion);
class FString GetEsportsTeamRegion(class UGFxData_ShopCatalogue_TA* EsportsTeamRow);
void eventConstruct();
};

// Class TAGame.ShopCatalogueCacheSave_TA
// 0x0010 (0x00C8 - 0x00D8)
class UShopCatalogueCacheSave_TA : public USaveObject_TA
{
public:
TArray<int32_t> ViewedShopIDs; // 0x00C8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ShopCatalogueCacheSave_TA");
}

return uClassPointer;
};

void SetShopViewed(int32_t ShopID);
bool IsShopViewed(int32_t ShopID);
};

// Class TAGame.ShopErrors_TA
// 0x0020 (0x0080 - 0x00A0)
class UShopErrors_TA : public UErrorList
{
public:
class UErrorType* ShopItemBadData; // 0x0080 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* ShopItemDiscountBadData; // 0x0088 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* ShopSyncedBadData; // 0x0090 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* ShopNotEnoughCurrency; // 0x0098 (0x0008)

```

[0x0000000000000002] (CPF\_Const)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ShopErrors_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.RPC_GetSpecialEventCurrencyRewards_TA  
// 0x0068 (0x00E8 - 0x0150)  
class URPC_GetSpecialEventCurrencyRewards_TA : public URPC_SpecialEventBase_TA  
{  
public:  
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)  
[0x0001000000400000] (CPF_NeedCtorLink)  
TArray<int32_t> EventIDs; // 0x0130 (0x0010)  
[0x0001000000400000] (CPF_NeedCtorLink)  
TArray<struct FSpecialEventStore> EventStores; // 0x0140 (0x0010)  
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.RPC_GetSpecialEventCurrencyRewards_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_GetSpecialEventCurrencyRewards_TA* SetPlayerID(struct FUniqueNetId InPlayerId);  
class URPC_GetSpecialEventCurrencyRewards_TA* SetEventIDs(TArray<int32_t> InEventIDs);  
};  
  
// Class TAGame.GFxData_SplitscreenMatch_TA  
// 0x0006 (0x019A - 0x01A0)  
class UGFxData_SplitscreenMatch_TA : public UGFxData_PrivateMatch_TA  
{  
public:  
  
public:
```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SplitscreenMatch_TA");
}

return uClassPointer;
};

void StartSearch();
};

// Class TAGame.GFxData_SupportACreator_TA
// 0x0038 (0x0098 - 0x00D0)
class UGFxData_SupportACreator_TA : public UGFxDataSingleton_X
{
public:
class FString CreatorCode; // 0x0098 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
uint64_t ExpirationDate; // 0x00A8 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
uint8_t SupportType; // 0x00B0 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
class UGFxModal_X* ProcessingModal; // 0x00B8 (0x0008)
[0x000100000002000] (CPFTransient)
TArray<class FString> CreatorIcons; // 0x00C0 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_SupportACreator_TA");
}

return uClassPointer;
};

void __GFxData_SupportACreator_TA__GetSupportedCreator_0x1(class
URPC_SupportACreator_Get_TA* RPC);
void __GFxData_SupportACreator_TA__OnClearSupportACreator_0x2(class URPC_X* RPC);
void __GFxData_SupportACreator_TA__OnClearSupportACreator_0x1(class
URPC_SupportACreator_Clear_TA* RPC);
void __GFxData_SupportACreator_TA__OnSetSupportACreator_0x2(class URPC_X* RPC);
void __GFxData_SupportACreator_TA__OnSetSupportACreator_0x1(class
URPC_SupportACreator_Set_TA* RPC);
void OnClearSupportACreatorFail();
void HandleClearSupportACreatorFail(class UError* Error);

```

```

void OnSetSupportACreatorFail();
void HandleSetSupportACreatorFail(class UError* Error);
void OnClearSupportACreatorSuccess();
void HandleClearSupportACreatorSuccess();
void OnSetSupportACreatorSuccess();
class FString GetIconName(uint8_t InSupportType);
void HandleSetSupportACreatorSuccess(class URPC_SupportACreator_Set_TA* RPC);
void OnSetSupportACreator(class FString InCreatorCode);
void OnClearSupportACreator();
void SetSupportACreator(class FString InCreatorCode, unsigned long bCanceled);
void OpenSupportACreator();
void GetSupportedCreator();
void ClearModal(class UGFxModal_X* Modal);
void HandlePsyNetLogin();
void eventOnShellSet();
};

// Class TAGame.RPC_SupportACreator_Get_TA
// 0x0028 (0x00E8 - 0x0110)
class URPC_SupportACreator_Get_TA : public URPC_X
{
public:
    class FString             CreatorCode;           // 0x00E8 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    uint64_t                  ExpirationTime;        // 0x00F8 (0x0008)
    [0x0000000000002000] (CPF_Transient)
    class FString             SupportType;          // 0x0100 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_SupportACreator_Get_TA");
        }

        return uClassPointer;
    };
};

// Class TAGame.RPC_SupportACreator_Clear_TA
// 0x0000 (0x00E8 - 0x00E8)
class URPC_SupportACreator_Clear_TA : public URPC_X
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_SupportACreator_Clear_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_SupportACreator_Set_TA
// 0x0030 (0x00E8 - 0x0118)
class URPC_SupportACreator_Set_TA : public URPC_X
{
public:
class FString CreatorCode; // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long Success : 1; // 0x00F8 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
uint64_t ExpirationTime; // 0x0100 (0x0008)
[0x0000000000002000] (CPF_Transient)
class FString SupportType; // 0x0108 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_SupportACreator_Set_TA");
}

return uClassPointer;
};

class URPC_SupportACreator_Set_TA* SetCreatorCode(class FString InCreatorCode);
};

// Class TAGame.SupportACreatorErrors_TA
// 0x0008 (0x0080 - 0x0088)
class USupportACreatorErrors_TA : public UErrorList
{
public:
class UErrorType* InvalidSupportACreatorCode; // 0x0080 (0x0008)
[0x00100000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SupportACreatorErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.TourBracketSync_TA
// 0x0068 (0x0060 - 0x00C8)
class UTourBracketSync_TA : public UObject
{
public:
class UOnlineGameTournaments_TA* Tournaments; // 0x0060
(0x0008) [0x0001800000000001] (CPF_Edit)
float PollIntervalSeconds; // 0x0068 (0x0004)
[0x0001000000000001] (CPF_Edit)
struct FUniqueNetId PlayerID; // 0x0070 (0x0048)
[0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTourSettings_TA* Settings; // 0x00B8 (0x0008)
[0x0001004000002000] (CPF_Transient)
class UAsyncTask* GetBracketTask; // 0x00C0 (0x0008)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourBracketSync_TA");
}

return uClassPointer;
};

void __TourBracketSync_TA__SendRequest_0x1(class UError* _);
void SendRequest();
void Deactivate();
void Start(struct FUniqueNetId InPlayerId, class UTourSettings_TA* InSettings);
};

// Class TAGame.GFxData_TourBracketGame_TA
// 0x0024 (0x0094 - 0x00B8)
class UGFxData_TourBracketGame_TA : public UGFxDataRow_X
{
public:
int32_t MatchID; // 0x0098 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t GameNum; // 0x009C (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)

```

```

int32_t           TeamScore0;          // 0x00A0 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t           TeamScore1;          // 0x00A4 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
int32_t           GameTimeSeconds;    // 0x00A8 (0x0004)
[0x0001000040000000] (CPF_EditInlineNotify)
unsigned long      bOverTime : 1;       // 0x00AC (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
uint64_t          WinnerTeam;         // 0x00B0 (0x0008)
[0x0001000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourBracketGame_TA");
}

return uClassPointer;
};

void SetGame(int32_t InMatchID, int32_t InGameNum, struct FTourMatchGame& MatchGame);
};

// Class TAGame.GFxData_TourBracketLeaderboard_TA
// 0x0040 (0x0098 - 0x00D8)
class UGFxData_TourBracketLeaderboard_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FTourPlacementInfo>      TeamGoals;          // 0x0098
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FTourPlacementInfo>      PlayerGoals;        // 0x00A8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FTourPlacementInfo>      PlayerAssists;     // 0x00B8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
TArray<struct FTourPlacementInfo>      PlayerSaves;       // 0x00C8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourBracketLeaderboard_TA");
}

return uClassPointer;
};

```

```

void Update(struct FTourLeaderboard& Leaderboard);
};

// Class TAGame.TournamentSettingsSave_TA
// 0x00B0 (0x00C8 - 0x0178)
class UTournamentSettingsSave_TA : public USaveObject_TA
{
public:
    struct FTourCreateSettings           TourCreateSettings;          // 0x00C8 (0x0038)
    [0x0001000000400000] (CPF_NeedCtorLink)
    struct FTourSearchSettings          TourSearchSettings;        // 0x0100 (0x0060)
    [0x0001000000400000] (CPF_NeedCtorLink)
    class UGameTags_TA*                GameTags;                  // 0x0160 (0x0008)
    [0x0001000000000000]
    class FString                      CheckInTeamName;         // 0x0168 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.TournamentSettingsSave_TA");
        }

        return uClassPointer;
    }

void SetTourCreateSettings(struct FCustomMatchSettings& MatchSettings);
    struct FCustomMatchSettings GetTourCreateSettings();
    void OnLoad();
};

// Class TAGame.TourCredentials_TA
// 0x0010 (0x0060 - 0x0070)
class UTourCredentials_TA : public UObject
{
public:
    TArray<struct FCredentialItem>      CredentialsMap;          // 0x0060 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.TourCredentials_TA");
        }

        return uClassPointer;
    }
}

```

```

};

bool Get(uint64_t TourID, struct FTourPrivateCredentials& Credentials);
void Set(uint64_t TourID, struct FTourPrivateCredentials Credentials);
};

// Class TAGame.GFxData_TourCheckInError_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_TourCheckInError_TA : public UGFxDataSingleton_X
{
public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x0098
(0x0008) [0x0001800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourCheckInError_TA");
}

return uClassPointer;
};

void EndUpdate();
void BeginUpdate(uint64_t TournamentID);
};

// Class TAGame.GFxData_TourConfig_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_TourConfig_TA : public UGFxDataSingleton_X
{
public:
class UTourConfig_TA*           Config;             // 0x0098 (0x0008)
[0x0001800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourConfig_TA");
}

return uClassPointer;
};

void HandleConfigChange();
};

```

```

// Class TAGame.GFxData_TourDependentGame_TA
// 0x0000 (0x00B8 - 0x00B8)
class UGFxData_TourDependentGame_TA : public UGFxData_TourBracketGame_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourDependentGame_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_TourDependentTeam_TA
// 0x0000 (0x00B8 - 0x00B8)
class UGFxData_TourDependentTeam_TA : public UGFxData_TourBracketTeam_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourDependentTeam_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_TourDetails_TA
// 0x0054 (0x0098 - 0x00EC)
class UGFxData_TourDetails_TA : public UGFxDataSingleton_X
{
public:
class UTourSettings_TA*           Settings;           // 0x0098 (0x0008)
[0x0001004000002000] (CPF_Transient)
class UOnlineProductStoreSet_TA*   OnlineProductSet;   // 0x00A0
(0x0008) [0x0001000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UTourDetailsSync_TA*         DetailsSync;       // 0x00A8 (0x0008)

```

```

[0x0001000000000000]
class UTourSettings_TA*           TourSettings;          // 0x00B0 (0x0008)
[0x0001000000000000]
class UOnlineGameTournaments_TA*   Tournaments;         // 0x00B8
(0x0008) [0x0001800000000001] (CPF_Edit)
TArray<struct FGFxTournamentReward> Rewards;           // 0x00C0
(0x0010) [0x0001000040400001] (CPF>Edit | CPF_NeedCtorLink | CPF>EditInlineNotify)
class UTexture*                  CreatorAvatar;        // 0x00D0 (0x0008)
[0x0001000040000000] (CPF>EditInlineNotify)
class FString                   CreatorPlatform;      // 0x00D8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
unsigned long                   bRegistered : 1;     // 0x00E8 (0x0004)
[0x0001000040000000] [0x00000001] (CPF>EditInlineNotify)
unsigned long                   bAdmin : 1;          // 0x00E8 (0x0004)
[0x0001000040000000] [0x00000002] (CPF>EditInlineNotify)
unsigned long                   bCanCheckIn : 1;    // 0x00E8 (0x0004)
[0x0001000040000000] [0x00000004] (CPF>EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourDetails_TA");
}

return uClassPointer;
};

uint64_t __GFxData_TourDetails_TA__LogTournamentNotFound_0x1(class UTourSettings_TA* T);
void __GFxData_TourDetails_TA__Update_0x1(struct FUniqueNetId InPlayerId, class UTexture* InAvatar, class FString OnlinePlayerName);
TArray<int32_t> GetSeriesRoundLengths();
void HandleSubscriptionChanged(class UTourSubscriptions_TA* Subscriptions, uint64_t TournamentID);
void UpdateCheckInWindow();
void HandleTourAdded(class UTourSettings_TA* NewSettings);
void HandleCheckInWindowChange(class UTourCheckInWindowWatcher_TA* Watcher, class UTourSettings_TA* InSettings);
void HandleRewardsComplete(uint64_t TournamentID, TArray<struct FTourReward> InRewards);
void HandleError(class UError* Error);
void MenuClosed();
bool Update(uint64_t TournamentID);
void LogTournamentNotFound(uint64_t TournamentID);
void SetProxy(class UObject* InProxyObject);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.TourDetailsSync_TA
// 0x0068 (0x0060 - 0x00C8)
class UTourDetailsSync_TA : public UObject

```

```

{
public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x0060
(0x0008) [0x0018000000000001] (CPF_Edit)
float                         PollIntervalSeconds;      // 0x0068 (0x0004)
[0x0010000000000001] (CPF_Edit)
struct FUniqueNetId                  PlayerID;            // 0x0070 (0x0048)
[0x001004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UTourSettings_TA*              Settings;            // 0x00B8 (0x0008)
[0x001004000002000] (CPF_Transient)
class UAsyncTask*                   GetBracketTask;       // 0x00C0 (0x0008)
[0x001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourDetailsSync_TA");
}

return uClassPointer;
};

void __TourDetailsSync_TA__SendRequest_0x1(class UError* _);
bool __TourDetailsSync_TA__UpdateTeamsRegistered_0x1(struct FTourTeam T);
void UpdateTeamsRegistered(struct FTourBracket& Bracket);
void HandleGetBracket(struct FTourBracket& Bracket);
void SendRequest();
void Deactivate();
void Start(struct FUniqueNetId InPlayerId, class UTourSettings_TA* InSettings);
};

// Class TAGame.TourRewards_TA
// 0x0038 (0x0060 - 0x0098)
class UTourRewards_TA : public UObject
{
public:
TArray<struct FRewardsResult>          Results;            // 0x0060 (0x0010)
[0x001004000400000] (CPF_NeedCtorLink)
TArray<struct FRewardsRequest>          Requests;           // 0x0070 (0x0010)
[0x001000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 __OnGetRewards__Delegate; // 0x0080
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.TourRewards_TA");
}

return uClassPointer;
};

void HandleGetRewards(uint64_t TournamentID, TArray<struct FTourReward> Rewards);
class UAsyncTask* Request(class UTourService_TA* Service, struct FUniqueNetId PlayerID,
uint64_t TournamentID, struct FScriptDelegate Callback);
void OnGetRewards(uint64_t TournamentID, TArray<struct FTourReward> Rewards);
};

// Class TAGame.GFxData_TourMatchmaking_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_TourMatchmaking_TA : public UGFxDataSingleton_X
{
public:
class UOnlineGameTourMatchmaking_TA* TourMatchmaking; // 0x0098
(0x0008) [0x0001800000000001] (CPF_Edit)
class UOnlineGameVersion_X* GameVersion; // 0x00A0
(0x0008) [0x0001800000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourMatchmaking_TA");
}

return uClassPointer;
};

void HandleMatchmakingError(class FString NewStatus);
void eventOnShellSet();
};

// Class TAGame.GFxData_TourSearchResult_TA
// 0x0000 (0x00B0 - 0x00B0)
class UGFxData_TourSearchResult_TA : public UGFxData_TourSubscription_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourSearchResult_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.RPC_TourGetPublic_TA
// 0x00C8 (0x00E8 - 0x01B0)
class URPC_TourGetPublic_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0001004000400000] (CPF_NeedCtorLink)
struct FTourSearchSettings     Search;            // 0x0130 (0x0060)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId>    TeamMembers;        // 0x0190 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<class UTourSettings_TA*> Tournaments;      // 0x01A0
(0x0010) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetPublic_TA");
}

return uClassPointer;
};

class URPC_TourGetPublic_TA* SetTeamMembers(TArray<struct FUniqueNetId>
InTeamMembers);
class URPC_TourGetPublic_TA* SetSettings(struct FTourSearchSettings& InSettings);
class URPC_TourGetPublic_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.GFxData_TourSpectate_TA
// 0x0048 (0x0098 - 0x00E0)
class UGFxData_TourSpectate_TA : public UGFxDataSingleton_X
{
public:
class UOnlineGameTournaments_TA*   Tournaments;       // 0x0098
(0x0008) [0x0001800000000001] (CPF_Edit)
class FString                      MatchNotAvailable; // 0x00A0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                      SearchingTitle;   // 0x00B0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString                      SearchingBody;    // 0x00C0 (0x0010)
[0x0001000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UAsyncTask*                  ActiveTask;       // 0x00D0 (0x0008)
[0x000100000002000] (CPF_Transient)

```

```
uint64_t PendingTourID; // 0x00D8 (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourSpectate_TA");
}

return uClassPointer;
};

void HandleJoinGameComplete(unsigned long bSuccess, class FString FailReason);
void __GFxData_TourSpectate_TA__StartJoin_0x1();
void __GFxData_TourSpectate_TA__HandleGetMatchServer_0x1();
class UOnlineGameJoinGame_X* GetJoinGame();
void HandleSearchError(class FString msg);
void HandleGetMatchServer(struct FServerReservationData Reservation);
void HandleError(class UError* Error);
void StartJoin(uint64_t TournamentID, int32_t MatchID);
};

// Class TAGame.TourMatchServerBrowser_TA
// 0x0020 (0x0060 - 0x0080)
class UTourMatchServerBrowser_TA : public UObject
{
public:
class UAsyncTask* GetMatchTask; // 0x0060 (0x0008)
[0x0001000000000000]
struct FScriptDelegate __OnGetMatchServer__Delegate; // 0x0068
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourMatchServerBrowser_TA");
}

return uClassPointer;
};

void ClearTask();
class UAsyncTask* GetMatchServer(struct FUniqueNetId PlayerID, uint64_t TournamentID,
int32_t MatchID, struct FScriptDelegate Callback);
void OnGetMatchServer(struct FServerReservationData Reservation);
};
```

```
// Class TAGame.GFxData_TourStatus_TA
// 0x0008 (0x0098 - 0x00A0)
class UGFxData_TourStatus_TA : public UGFxDataSingleton_X
{
public:
    class UTourStatus_TA* Status; // 0x0098 (0x0008)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxData_TourStatus_TA");
        }

        return uClassPointer;
    }

    void eventOnShellSet();
};

// Class TAGame.RPC_TourCancelCreated_TA
// 0x0050 (0x00E8 - 0x0138)
class URPC_TourCancelCreated_TA : public URPC_X
{
public:
    struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    uint64_t TournamentID; // 0x0130 (0x0008)
    [0x0001004000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourCancelCreated_TA");
        }

        return uClassPointer;
    }

    class URPC_TourCancelCreated_TA* SetTournamentID(uint64_t InTournamentID);
    class URPC_TourCancelCreated_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.TrainingConfig_TA
// 0x0008 (0x0078 - 0x0080)
```

```

class UTrainingConfig_TA : public UOnlineConfig_X
{
public:
int32_t           MaxHistoryItems;          // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long      bCanShowHistoryTab : 1;    // 0x007C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long      bCanShowProgressV2 : 1;    // 0x007C (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long      bEnableTrainingUIV2 : 1;   // 0x007C (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.TrainingProgressTracker_TA
// 0x0048 (0x0060 - 0x00A8)
class UTrainingProgressTracker_TA : public UObject
{
public:
class AGameEvent_TrainingEditor_TA*      TrainingEditor;        // 0x0060
(0x0008) [0x0000000000002000] (CPF_Transient)
class UTrainingPackProgress_TA*          CurrentProgress;     // 0x0068
(0x0008) [0x0000000000002000] (CPF_Transient)
class UTrainingProgressFactory_TA*       ProgressFactory;     // 0x0070
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate                 __EventProgressSet__Delegate; // 0x0078
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate                 __ValidateRound__Delegate; // 0x0090
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingProgressTracker_TA");
}

return uClassPointer;
}

```

};

```
void SetCurrentProgress(class UTrainingPackProgress_TA* NewProgress);
void SetProgressSave(class UTrainingProgressSave_TA* InProgressSave, struct FScriptDelegate
ValidateRoundCallback);
void HandleRoundAttempted(int32_t RoundNumber, uint8_t NewAttempt);
void HandlePlaytestStarted(class AGameEvent_TrainingEditor_TA* InTrainingEditor, struct
FScriptDelegate ValidateRoundCallback, uint64_t InCurrentTime);
void Init(class AGameEvent_TrainingEditor_TA* InTrainingEditor);
bool ValidateRound(struct FEditorRoundData RoundData);
void EventProgressSet(class UTrainingPackProgress_TA* NewProgress, class
UTrainingPackProgress_TA* OldProgress);
};
```

```
// Class TAGame.GFxData_TrainingMode_TA
// 0x0184 (0x0094 - 0x0218)
class UGFxData_TrainingMode_TA : public UGFxDataRow_X
{
public:
class UTrainingConfig_TA*           TrainingModeConfig;          // 0x0098 (0x0008)
[0x0000800000000000]
uint8_t                           Status;                  // 0x00A0 (0x0001)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class FString                     SanitizedTrainingModeName; // 0x00A8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString                     SanitizedCreatorName;    // 0x00B8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString                     Tags;                   // 0x00C8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
unsigned long                      bOwned : 1;            // 0x00D8 (0x0004)
[0x0000000040002000] [0x00000001] (CPF_Transient | CPF_EditInlineNotify)
unsigned long                      bFavorited : 1;       // 0x00D8 (0x0004)
[0x0000000040002000] [0x00000002] (CPF_Transient | CPF_EditInlineNotify)
unsigned long                      bOutOfDate : 1;        // 0x00D8 (0x0004)
[0x0000000040002000] [0x00000004] (CPF_Transient | CPF_EditInlineNotify)
class FString                     PublishedTime;         // 0x00E0 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString                     UpdatedAtTime;        // 0x00F0 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
int32_t                           ShotsCompleted;        // 0x0100 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class FString                     CreatorPlatform;       // 0x0108 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF_EditInlineNotify)
class UTexture*                  CreatorAvatar;         // 0x0118 (0x0008)
[0x0000000040002000] (CPF_Transient | CPF_EditInlineNotify)
class FString                     SaveDataFilename;      // 0x0120 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                     SavingFilename;        // 0x0130 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString                     AutoplayName;         // 0x0140 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class USaveData_GameEditor_Training_TA* SaveData;          // 0x0150
(0x0008) [0x0000000000002000] (CPF_Transient)
class FString                     OpenTrainingMapBaseCmd; // 0x0158 (0x0010)
```

```

[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
class FString OutOfDateMessage; // 0x0168 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString DownloadingTrainingModeTitleMessage; // 0x0178
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString DownloadingTrainingModeBodyMessage; // 0x0188
(0x0010) [0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UGFxModal_X* ModalProcessing; // 0x0198 (0x0008)
[0x000000000002000] (CPF_Transient)
struct FScriptDelegate __EventTrainingModeDownloadPlay__Delegate; //
0x01A0 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTrainingModeFavorited__Delegate; // 0x01B8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTrainingModeLoaded__Delegate; // 0x01D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTrainingModeSaved__Delegate; // 0x01E8
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventTrainingModeWordFilterError__Delegate; //
0x0200 (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TrainingMode_TA");
}

return uClassPointer;
};

void __GFxData_TrainingMode_TA__HandleSaveError_0x1(class UGFxModal_X* _);
bool __GFxData_TrainingMode_TA__UpdateNumRoundsAndShotsCompleted_0x1(struct
FTrainingRoundProgress RoundProgress);
bool CanShowProgressV2();
void UpdateNumRoundsAndShotsCompleted(class UTrainingEditorData_TA* InTrainingData);
void UpdateOutOfDate(uint64_t OnlineUpdatedAt);
void HandleUnpublish(class URPC_TrainingEditor_DeleteTrainingData_TA* RPC);
void OnDelete(uint8_t CurrentTab);
void ForceDelete(class FString FilePath);
void HandleDispose();
void AddToFavorites();
void LoadFile(class FString Filename, struct FName inMapName, unsigned long bInPlaytest);
void Play(struct FName inMapName, unsigned long bInPlaytest);
void HandleDownloadAndPlayResults(class URPC_TrainingEditor_GetTrainingData_TA* RPC);
void HandleDownload(class URPC_TrainingEditor_GetTrainingData_TA* RPC);
void HandleUpdateTrainingData(class URPC_TrainingEditor_UpdateTrainingData_TA* RPC);
void HandleAddTrainingData(class URPC_TrainingEditor_AddTrainingData_TA* RPC);
void HandleSubmit(class UPrivilegeCheck_X* PrivilegeCheck);
void Submit();
void HandleCreatorNameSanitized(class FString DesiredName, class FString SanitizedName);
void HandleTrainingModeNameSanitized(class FString DesiredName, class FString

```

```

SanitizedName);
void OnDataLoaded(struct FLoadObjectResult Result);
void HandleSanitizeError(class FString OriginalPhrase, class UError* Error);
void SetTrainingModeFromSave(class FString Filename);
void HandleGetAvatar(struct FUniqueNetId InPlayerId, class UTexture* InAvatar, class FString OnlinePlayerName);
void SetTrainingMode(class UTrainingEditorData_TA* InTrainingData);
void UpdateUITimes();
void SetTrainingModeFromRPC(class UTrainingEditorData_TA* ResultData);
void HandleAutoplay(struct FSaveObjectResult Result);
void HandleSaveResult(struct FSaveObjectResult Result);
void OnRetrySaveSaveData(class UGFxModal_X* Modal);
void HandleSaveError(struct FSaveObjectResult Result);
void SaveTrainingMode(class FString Filename, class UTrainingEditorData_TA* NewTrainingData,
unsigned long bInOwned, unsigned long bAutoPlay);
void DisplayError(class UError* Error);
class UTrainingEditorMetrics_TA* GetTrainingMetrics();
void eventOnShellSet();
void EventTrainingModeWordFilterError(class UGFxData_TrainingMode_TA* OutTrainingMode,
class UError* Error);
void EventTrainingModeSaved(class UGFxData_TrainingMode_TA* OutTrainingMode, unsigned long bSuccess);
void EventTrainingModeLoaded(class UGFxData_TrainingMode_TA* OutTrainingMode, unsigned long bSuccess);
void EventTrainingModeFavorited(class UGFxData_TrainingMode_TA* OutTrainingMode,
unsigned long bAdded);
void EventTrainingModeDownloadPlay(class UGFxData_TrainingMode_TA* OutTrainingMode,
class FString Code, class FString Filename);
};


```

```

// Class TAGame.GFxData_TrainingModeBrowser_TA
// 0x00D0 (0x0098 - 0x0168)
class UGFxData_TrainingModeBrowser_TA : public UGFxDataSingleton_X
{
public:
class UTrainingConfig_TA*           TrainingBrowserConfig;          // 0x0098
(0x0008) [0x0000800000000000]
struct FSearchCacheData            CachedOnlineSearchResults[0x2];    // 0x00A0
(0x0030) [0x0000000000400000] (CPF_NeedCtorLink)
class URPC_X*                      PendingRPC;                  // 0x00D0 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                             CachedSearchDuration;        // 0x00D8 (0x0004)
[0x0000000000000002] (CPF_Const)
float                             CachedFavoriteTimeoutTime; // 0x00DC (0x0004)
[0x0000000000002000] (CPF_Transient)
TArray<struct FFavoritedFileMetaData> FavoritedFilesMetaData;      // 0x00E0
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
unsigned long                     bFavoritesLoaded : 1;          // 0x00F0 (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)
unsigned long                     bErrorModalOpen : 1;          // 0x00F0 (0x0004)
[0x0000000000002000] [0x00000002] (CPF_Transient)
float                           CachedFavoriteDuration;       // 0x00F4 (0x0004)
[0x0000000000000002] (CPF_Const)
TArray<struct FCachedDownloadedFile> CachedDownloadedFiles;        //

```

```
0x00F8 (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class FString> LoadingFiles; // 0x0108 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class FString DownloadedFilename; // 0x0118 (0x0010)
[0x0000000000400002] (CPF_Const | CPF_NeedCtorLink)
int32_t MaxCachedDownloadedFiles; // 0x0128 (0x0004)
[0x0000000000000002] (CPF_Const)
class USaveData_GameEditor_Training_TA* TrainingSaveData; // 0x0130
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate __EventFavoritesLoaded__Delegate; // 0x0138
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)
struct FScriptDelegate __EventAllTrainingFilesLoaded__Delegate; // 0x0150
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TrainingModeBrowser_TA");
}

return uClassPointer;
};

class FString __GFxDATA_TrainingModeBrowser_TA__GetTrainingHistory_0x2(class
UTrainingPackProgress_TA* Progress);
int32_t __GFxDATA_TrainingModeBrowser_TA__GetTrainingHistory_0x1(class
UTrainingPackProgress_TA* L, class UTrainingPackProgress_TA* R);
void InitializeTrainingModeFromMetadata(class UTrainingEditorData_TA* InTrainingData);
void HandleHistoryResults(class URPC_TrainingEditor_GetTrainingMetadata_TA* RPC);
void GetTrainingHistory();
bool CanShowHistoryTab();
bool CanUploadCustomPack();
class UGFxDATA_TrainingMode_TA* DebugGetTrainingModeByIdx(int32_t Index);
class UGFxDATA_TrainingMode_TA* DebugGetTrainingModeByName(class FString TM_Name);
class FString GetDownloadedFilename();
void DeleteTrainingMode(int32_t InRowIndex, uint8_t CurrentTab);
void DeleteTrainingModeData(class UGFxDATA_TrainingMode_TA* TrainingMode, uint8_t
CurrentTab);
void OnTrainingModesReady();
void SetTrainingFiles(TArray<class FString> Files, unsigned long bLoadingFavorites);
void HandleHeaders(TArray<class FString> Files, uint8_t Result);
void HandleFavoritesLoaded(class UGFxDATA_TrainingModeBrowser_TA* B);
void GetLocalTrainingFiles(unsigned long bLoadingFavorites);
void HandleTrainingModeDownloadPlay(class UGFxDATA_TrainingMode_TA* OutTrainingMode,
class FString InCode, class FString Filename);
void HandleTrainingModeFavorited(class UGFxDATA_TrainingMode_TA* OutTrainingMode,
unsigned long bAdded);
int32_t GetFavoriteIndex(class FString Id);
bool IsFavorited(class FString InGuid);
void HandleBrowseResults(class URPC_TrainingEditor_BrowseTrainingData_TA* RPC);
```

```

void HandleSearchByCode(class URPC_TrainingEditor_GetTrainingData_TA* RPC);
void OnSearchCodeResults(unsigned long bSuccess);
void HandleOnlineHeaders(TArray<class FString> LocalFiles, uint8_t Result, unsigned long
bFeaturedOnly);
void HandleLoadFeaturedHeaders(TArray<class FString> LocalFiles, uint8_t Result);
void HandleLoadAllPublicHeaders(TArray<class FString> LocalFiles, uint8_t Result);
bool LoadIfFavorited(class UGFxData_TrainingMode_TA* GFxTrainingData, struct FGuid InGuid);
void RepopulateFromCache(unsigned long bFeaturedOnly);
void GetSearchResults(class FString Code, unsigned long bForceRefresh, unsigned long
bFeaturedOnly);
class APlayerControllerBase_TA* GetPlayerController();
void HandleAllFavoritesLoaded(class UGFxData_TrainingModeBrowser_TA*
TrainingModeBrowser);
void ResetTrainingList();
void HandleDownloadResults(class URPC_TrainingEditor_GetTrainingData_TA* RPC);
void HandleListPlayerTrainingData(class URPC_TrainingEditor_ListPlayerTrainingData_TA* RPC);
void SyncTrainingModes();
void HandleListPlayerTrainingDataTimestamp(class
URPC_TrainingEditor_ListPlayerTrainingDataTimestamp_TA* RPC);
void EvaluateUpdateAtTimes();
void SetFavoriteUpdatedAtTime(class FString InCode, uint64_t InUpdatedAt);
void SyncFavoritedModes();
void HandleTrainingModeWordFilterError(class UGFxData_TrainingMode_TA* OutTrainingMode,
class UError* Error);
void HandleLocalTrainingModeLoaded(class UGFxData_TrainingMode_TA* OutTrainingMode,
unsigned long bSuccess);
void HandleTrainingModeLoaded(class UGFxData_TrainingMode_TA* OutTrainingMode,
unsigned long bSuccess);
void HandleDownloadedTrainingModeSaved(class UGFxData_TrainingMode_TA*
OutTrainingMode, unsigned long bSuccess);
void HandleLocalTrainingModeCreated(class UGFxData_TrainingMode_TA* OutTrainingMode,
unsigned long bSuccess);
void HandleLocalTrainingModeSaved(class UGFxData_TrainingMode_TA* OutTrainingMode,
unsigned long bSuccess);
void CreateNew(class FString NewName, uint8_t NewType, uint8_t NewDifficulty, struct FName
MapName, TArray<int32_t> Tags);
void HandleErrorClosed(class UGFxModal_X* Modal);
void DisplayError(class UError* Error);
class UTrainingEditorMetrics_TA* GetTrainingMetrics();
void NotifyWhenFavoritesLoaded(struct FScriptDelegate Callback);
void HandleFavorites(TArray<class FString> Files, uint8_t Result);
void HandlePsyNetLogin(class UOnlinePlayer_X* OnlinePlayer);
void OnShellSet();
void EventAllTrainingFilesLoaded(class UGFxData_TrainingModeBrowser_TA* Browser);
void EventFavoritesLoaded(class UGFxData_TrainingModeBrowser_TA* Browser);
};


```

```

// Class TAGame.RPC_TrainingEditor_AddTrainingData_TA
// 0x0018 (0x00E8 - 0x0100)
class URPC_TrainingEditor_AddTrainingData_TA : public URPC_X
{
public:
class UTrainingEditorData_TA*           TrainingData;          // 0x00E8 (0x0008)
[0x0000000000000000]

```

```
class FString                         Code;           // 0x00F0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TrainingEditor_AddTrainingData_TA");
}

return uClassPointer;
};

class URPC_TrainingEditor_AddTrainingData_TA* SetTrainingData(class
UTrainingEditorData_TA*& InTrainingData);
};

// Class TAGame.RPC_TrainingEditor_UpdateTrainingData_TA
// 0x0008 (0x00E8 - 0x00F0)
class URPC_TrainingEditor_UpdateTrainingData_TA : public URPC_X
{
public:
class UTrainingEditorData_TA*          TrainingData;      // 0x00E8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_TrainingEditor_UpdateTrainingData_TA");
}

return uClassPointer;
};

class URPC_TrainingEditor_UpdateTrainingData_TA* SetTrainingData(class
UTrainingEditorData_TA*& InTrainingData);
};

// Class TAGame.RPC_TrainingEditor_GetTrainingData_TA
// 0x0018 (0x00E8 - 0x0100)
class URPC_TrainingEditor_GetTrainingData_TA : public URPC_X
{
public:
class FString                         Code;           // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class UTrainingEditorData_TA*          TrainingData;      // 0x00F8 (0x0008)
```

[0x0000000000000000] (CPF\_Transient)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.RPC_TrainingEditor_GetTrainingData_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TrainingEditor_GetTrainingData_TA* SetTrainingModelID(class FString  
InTrainingModeCode);  
};  
  
// Class TAGame.GFxData_TrainingModeConstants_TA  
// 0x0038 (0x0098 - 0x00D0)  
class UGFxData_TrainingModeConstants_TA : public UGFxDataSingleton_X  
{  
public:  
TArray<struct FTrainingModeDifficulty> Difficulties; // 0x0098 (0x0010)  
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)  
TArray<struct FTrainingModeType> Types; // 0x00A8 (0x0010)  
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)  
TArray<struct FTrainingModeTag> Tags; // 0x00B8 (0x0010)  
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)  
struct FName DefaultMap; // 0x00C8 (0x0008)  
[0x0000000040000000] (CPF>EditInlineNotify)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TrainingModeConstants_TA");  
}  
  
return uClassPointer;  
};  
  
static struct FName GetValidatedMap(struct FName MapName);  
void InitTags();  
void OnShellSet();  
};  
  
// Class TAGame.RPC_TrainingEditor_DeleteTrainingData_TA  
// 0x0010 (0x00E8 - 0x00F8)  
class URPC_TrainingEditor_DeleteTrainingData_TA : public URPC_X
```

```
{  
public:  
class FString Code; // 0x00E8 (0x0010)  
[0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.RPC_TrainingEditor_DeleteTrainingData_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TrainingEditor_DeleteTrainingData_TA* SetTrainingModeCode(class FString  
InCode);  
};  
  
// Class TAGame.RPC_TrainingEditor_ListPlayerTrainingDataTimestamp_TA  
// 0x0020 (0x00E8 - 0x0108)  
class URPC_TrainingEditor_ListPlayerTrainingDataTimestamp_TA : public URPC_X  
{  
public:  
TArray<class FString> Codes; // 0x00E8 (0x0010)  
[0x0000000000400000] (CPF_NeedCtorLink)  
TArray<struct FCodeTimePair> TrainingData; // 0x00F8 (0x0010)  
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.RPC_TrainingEditor_ListPlayerTrainingDataTimestamp_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TrainingEditor_ListPlayerTrainingDataTimestamp_TA* SetCodes(TArray<class  
FString>& InCodes);  
};  
  
// Class TAGame.RPC_TrainingEditor_ListPlayerTrainingData_TA  
// 0x0010 (0x00E8 - 0x00F8)  
class URPC_TrainingEditor_ListPlayerTrainingData_TA : public URPC_X
```

```
{  
public:  
TArray<class FString> Codes; // 0x00E8 (0x0010)  
[0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.RPC_TrainingEditor_ListPlayerTrainingData_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.RPC_TrainingEditor_BrowseTrainingData_TA  
// 0x0018 (0x00E8 - 0x0100)  
class URPC_TrainingEditor_BrowseTrainingData_TA : public URPC_X  
{  
public:  
unsigned long bFeaturedOnly : 1; // 0x00E8 (0x0004)  
[0x0000000000000000] [0x00000001]  
TArray<class UTrainingEditorData_TA*> TrainingData; // 0x00F0  
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.RPC_TrainingEditor_BrowseTrainingData_TA");  
}  
  
return uClassPointer;  
};  
  
class URPC_TrainingEditor_BrowseTrainingData_TA* SetFeaturedOnly(unsigned long  
bInFeaturedOnly);  
};  
  
// Class TAGame.RPC_TrainingEditor_GetTrainingMetadata_TA  
// 0x0020 (0x00E8 - 0x0108)  
class URPC_TrainingEditor_GetTrainingMetadata_TA : public URPC_X  
{  
public:
```

```

TArray<class FString>          Codes;           // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<class UTrainingEditorData_TA*>   TrainingData;        // 0x00F8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_TrainingEditor_GetTrainingMetadata_TA");
}

return uClassPointer;
};

class URPC_TrainingEditor_GetTrainingMetadata_TA* SetCodes(TArray<class FString>&
InCodes);
};

// Class TAGame.GFxData_TrainingModeEditor_TA
// 0x0060 (0x0098 - 0x00F8)
class UGFxData_TrainingModeEditor_TA : public UGFxDataSingleton_X
{
public:
class AGameEvent_TrainingEditor_TA*      MyGameEvent;       // 0x0098
(0x0008) [0x0000000000002000] (CPF_Transient)
class FString                         SanitizedTrainingModeName; // 0x00A0 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FEditorRoundData>        RoundData;         // 0x00B0 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
int32_t                                CurrentRound;      // 0x00C0 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
TArray<struct FTrainingModeTag>        Tags;             // 0x00C8 (0x0010)
[0x0000000040402000] (CPF_Transient | CPF_NeedCtorLink | CPF>EditInlineNotify)
uint8_t                                 Editing;           // 0x00D8 (0x0001)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
float                                  BallStartSpeed;    // 0x00DC (0x0004)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
float                                  BallScreenPosX;   // 0x00E0 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
float                                  BallScreenPosY;   // 0x00E4 (0x0004)
[0x0000000040002000] (CPF_Transient | CPF>EditInlineNotify)
unsigned long                          bOwned : 1;        // 0x00E8 (0x0004)
[0x0000000040002000] [0x00000001] (CPF_Transient | CPF>EditInlineNotify)
unsigned long                          bNoEditor : 1;     // 0x00E8 (0x0004)
[0x0000000040002000] [0x00000002] (CPF_Transient | CPF>EditInlineNotify)
unsigned long                          bUnsavedChanges : 1; // 0x00E8 (0x0004)
[0x0000000040002000] [0x00000004] (CPF_Transient | CPF>EditInlineNotify)
class UGFxModal_X*                   ModalProcessing;    // 0x00F0 (0x0008)
[0x0000000000000000] (CPF_Transient)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_TrainingModeEditor_TA");
}

return uClassPointer;
};

void Save();
void HandleSaveResult(unsigned long bSuccess);
void HandleUnsavedChanges(unsigned long bInUnsavedChanges);
void DuplicateRound(int32_t Index);
void ReorderRound(int32_t FromIndex, int32_t ToIndex);
void StopEditingRound();
void EditCurrentRound();
void RestartSequence();
void TestSequence();
void TestCurrentRound();
void CreateRound();
void HandleRoundTimeChanged(class AGameEvent_TrainingEditor_TA* InGameEvent);
void HandleRoundChanged(class AGameEvent_TrainingEditor_TA* InGameEvent);
void SetGameEvent(class AGameEvent_TrainingEditor_TA* InGameEvent);
void HandleNameSanitized(class FString DesiredName, class FString SanitizedName);
void UpdateTrainingData();
void SetRoundTimeLimit(int32_t Time);
void DeleteRound(int32_t RoundToDelete);
void SetActiveRound(int32_t NewRound);
void SetMetaData(class FString NewName, uint8_t NewType, uint8_t NewDifficulty,
TArray<int32_t> NewTags);
void HandlePRIReceived(class APlayerController_X* PC_X);
class APlayerController_TA* GetPlayerController();
void OnShellSet();
};

// Class TAGame.GFxData_UIConfig_TA
// 0x0004 (0x0098 - 0x009C)
class UGFxData_UIConfig_TA : public UGFxDataSingleton_X
{
public:
unsigned long          bAprilFoolsRankedIcons : 1;           // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
unsigned long          bAprilFoolsPlaylistImages : 1;         // 0x0098 (0x0004)
[0x0000000040000000] [0x00000002] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_UIConfig_TA");
}

return uClassPointer;
};

uint8_t FindXPTier(int32_t XPAmount);
void EndAprilFools();
void StartAprilFools();
void HandleAprilConfigRemoved(class UAprilConfig_X* Config);
void HandleAprilConfig(class UAprilConfig_X* Config);
void HandleUIConfig(class UUIConfig_TA* Config);
void eventOnRemoved();
void eventOnShellSet();
};

// Class TAGame.GFxData_UserBugReport_TA
// 0x0078 (0x0098 - 0x0110)
class UGFxData_UserBugReport_TA : public UGFxDataSingleton_X
{
public:
class UUserBugReportConfig_TA*          Config;           // 0x0098 (0x0008)
[0x0001800000000000]
unsigned long              bEnabled : 1;           // 0x00A0 (0x0004)
[0x0001000040000000] [0x00000001] (CPF_EditInlineNotify)
TArray<struct FGFxUserBugReportCategory>   Categories;      // 0x00A8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FGFxUserBugReportCategory>   SelectedCategories; // 0x00B8
(0x0010) [0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString             UserMessage;     // 0x00C8 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
uint8_t                  Status;         // 0x00D8 (0x0001)
[0x0001000040000000] (CPF>EditInlineNotify)
class FString             SubmitError;    // 0x00E0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class FString             ConfirmationCode; // 0x00F0 (0x0010)
[0x0001000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
class UUserBugReportGeneratedData_TA*    PendingData;    // 0x0100
(0x0008) [0x0001000000000000]
class UTAsyncResult__UserBugReportUploadResult_TA* UploadTask; // 0x0108 (0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_UserBugReport_TA");
}
}

```

```

return uClassPointer;
};

struct FName __GFxData_UserBugReport_TA__SubmitReport_0x1(struct
FGFxUserBugReportCategory C);
void HandleSubmitReport(class UUserBugReportUploadResult_TA* Result, class UError* Error);
void SubmitReport();
void SetUserMessage(class FString Message);
void SetUserData(struct FName Category, struct FName SubCategory, class FString Message);
void Reset();
void CancelReport();
void GenerateReport();
struct FGFxUserBugReportCategory CreateGFxCategories(struct FUserBugReportCategory
Category);
void PopSelectedCategory();
void SelectCategory(struct FName Category);
void SetCategory(struct FName Category);
void ResetCategories();
void HandleConfigChanged();
void eventOnShellSet();
};

// Class TAGame.UserBugReportUploader_TA
// 0x0030 (0x0060 - 0x0090)
class UUserBugReportUploader_TA : public UObject
{
public:
class UPsyNetConnection_X*           Connection;           // 0x0060 (0x0008)
[0x0001000000000000]
class UUserBugReportBulkData_TA*     BulkData;           // 0x0068 (0x0008)
[0x0001000000000000]
class UTAsyncResult__UserBugReportUploadResult_TA* SubmitTask;      //
0x0070 (0x0008) [0x0001000000000000]
class URPC_SubmitUserBugReport_TA*   SubmitReportRPC;    // 0x0078
(0x0008) [0x0001000000000000]
class UUserBugReportUploadResult_TA* Result;             // 0x0080
(0x0008) [0x0001000000000000]
class UWebRequest_X*                UploadReportWebRequest; // 0x0088
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportUploader_TA");
}

return uClassPointer;
};

```

```

void HandleUploadComplete(class UWebRequest_X* Request);
void HandleSubmitReport(class URPC_SubmitUserBugReport_TA* RPC);
void Dispose();
void HandleTaskDisposed();
void SendRPC();
void HandleDescriptionFilter(struct FWordFilterResult FilterResult);
class UTAsyncResult__UserBugReportUploadResult_TA* Start(class UPsyNetConnection_X*
InConnection, class UUserBugReportSubmissionData_TA* UserData, class
UUserBugReportGeneratedData_TA* GeneratedData);
};

// Class TAGame.UserBugReportSubmissionData_TA
// 0x0020 (0x0060 - 0x0080)
class UUserBugReportSubmissionData_TA : public UObject
{
public:
TArray<struct FName> Categories; // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString UserMessage; // 0x0070 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportSubmissionData_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxData_ViewMetrics_TA
// 0x0010 (0x0098 - 0x00A8)
class UGFxData_ViewMetrics_TA : public UGFxDataSingleton_X
{
public:
unsigned long bRecordingMetrics : 1; // 0x0098 (0x0004)
[0x0000000040000000] [0x00000001] (CPF_EditInlineNotify)
int32_t ActiveViewID; // 0x009C (0x0004)
[0x0000000040000000] (CPF_EditInlineNotify)
class UViewMetricsConfig_TA* Config; // 0x00A0 (0x0008)
[0x0000800000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_ViewMetrics_TA");
}

return uClassPointer;
};

class UViewMetrics_TA* GetMetricObject();
void EndMetrics();
void AddInteraction(int32_t InteractableID, struct FName InteractableDescriptor);
void SetActiveView(int32_t ViewId);
void StartMetrics(int32_t SourceID, struct FName ViewDescriptor);
void eventOnRemoved();
};

// Class TAGame.ViewMetricsConfig_TA
// 0x0010 (0x0078 - 0x0088)
class UViewMetricsConfig_TA : public UOnlineConfig_X
{
public:
TArray<struct FName> DisabledViews; // 0x0078 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ViewMetricsConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.ViewMetrics_TA
// 0x001C (0x0080 - 0x009C)
class UViewMetrics_TA : public UMetricsGroup_X
{
public:
TArray<struct F ViewData> Metrics; // 0x0080 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
int32_t SourceID; // 0x0090 (0x0004)
[0x0000000000000000]
struct FName Descriptor; // 0x0094 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ViewMetrics_TA");
}

return uClassPointer;
};

void PrintViewMetricsData();
void MenulInteractionSessionCompleted(int32_t ViewSourceID, struct FName ViewDescriptor,
TArray<struct FViewData> ViewMetrics);
void AddInteraction(int32_t InteractableID, struct FName InteractableDescriptor);
void ViewChanged(int32_t ExitedViewID, int32_t EnteredViewID);
void StartMetrics(int32_t InSourceID, struct FName ViewDescriptor);
};

// Class TAGame.GFxData_Vote_TA
// 0x0024 (0x0094 - 0x00B8)
class UGFxData_Vote_TA : public UGFxDataRow_X
{
public:
    uint8_t           Subject;           // 0x0098 (0x0001)
    [0x0000000040000000] (CPF_EditInlineNotify)
    uint8_t           VoteStatus;        // 0x0099 (0x0001)
    [0x0000000040000000] (CPF_EditInlineNotify)
    uint8_t           LocalVoteStatus;   // 0x009A (0x0001)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t          YesVotes;          // 0x009C (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t          NoVotes;           // 0x00A0 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t          RequiredVotes;     // 0x00A4 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    int32_t          TimeRemaining;     // 0x00A8 (0x0004)
    [0x0000000040000000] (CPF_EditInlineNotify)
    class AVoteActor_TA*      VoteActor; // 0x00B0 (0x0008)
    [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Vote_TA");
}

return uClassPointer;
};

void SetVoteStatus(uint8_t Status);
void ClearVote();

```

```

void VoteNo();
void VoteYes();
void HandleDestroyed(class AVoteActor_TA* InVoteActor);
void HandleFinished(class AVoteActor_TA* InVoteActor);
void UpdateData(class AVoteActor_TA* InVoteActor);
void eventOnRemoved();
void SetVoteActor(class AVoteActor_TA* InVoteActor);
};

// Class TAGame.GFxData_WebBrowser_TA
// 0x0028 (0x0098 - 0x00C0)
class UGFxData_WebBrowser_TA : public UGFxDataSingleton_X
{
public:
class FString ActiveURL; // 0x0098 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class FString ActivePageTitle; // 0x00A8 (0x0010)
[0x0000000040400000] (CPF_NeedCtorLink | CPF_EditInlineNotify)
class UUrlConfig_X* UrlConfig; // 0x00B8 (0x0008)
[0x0000800000000001] (CPF>Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_WebBrowser_TA");
}

return uClassPointer;
};

void __GFxData_WebBrowser_TA__OnShellSet_0x1(class FString _);
void HandleSwitchingMap();
void HandleBrowserLoaded(class FString PageTitle, class FString URL, uint64_t LoadTimeInSeconds);
void HandleBrowserClosed();
void HandleBrowserOpened(class FString InURL);
void CloseBrowser();
void OpenBrowserToURL(class FString URL, int32_t BrowserWidth, int32_t BrowserHeight);
void eventOnShellSet();
};

// Class TAGame.GFxData_Workshop_TA
// 0x0030 (0x0098 - 0x00C8)
class UGFxData_Workshop_TA : public UGFxDataSingleton_X
{
public:
TArray<struct FDownloadedWorkshopData> Items; // 0x0098
(0x0010) [0x0000000040400000] (CPF_NeedCtorLink | CPF>EditInlineNotify)
TArray<struct FUniqueNetId> RetrievedNetIds; // 0x00A8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

```

```
class FString ItemRemoved; // 0x00B8 (0x0010)
[0x0000000000408002] (CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_Workshop_TA");
}

return uClassPointer;
};

struct FUniqueNetId __GFxData_Workshop_TA__HandleWorkshopData_0x1(struct
FDownloadedWorkshopData Item);
void OnAvatarPersonaHandler(struct FUniqueNetId PlayerNetId, class UTexture* Avatar, class
FString OnlinePlayerName);
void OnEventImageDownloaded(struct FOnlineImageDownload Image);
void HandleWorkshopData(unsigned long bSuccess, TArray<struct
FDownloadedWorkshopData>& WorkshopItems);
void LoadWorkshopMap(class FString& FullPath);
void LoadAllWorkshopMapData();
};

// Class TAGame.GFxData_XP_TA
// 0x0000 (0x0098 - 0x0098)
class UGFxData_XP_TA : public UGFxDataSingleton_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxData_XP_TA");
}

return uClassPointer;
};

void HandleSaveDataLoaded(class UClientXPSave_TA* XPSave);
void eventOnShellSet();
};

// Class TAGame.GRI_TA
// 0x0018 (0x06A8 - 0x06C0)
class AGRI_TA : public AGRI_X
{
```

```
public:
unsigned long          bHasTalkers : 1;           // 0x06A8 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
class FString          NewDedicatedServerIP;      // 0x06B0 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GRI_TA");
}

return uClassPointer;
};

void SetVoiceChatActive(unsigned long bActive);
void HandleTalkingStatusChange(class UOnlineGameVoice_X* OnlineGameVoice, class APlayerReplicationInfo* PRI, unsigned long bTalking);
class FString GetTimeString(int32_t Seconds, unsigned long bCountDown, unsigned long bShowMinutes, float MS, unsigned long bShowMS);
void eventDestroyed();
void SetDefaultScriptedColors();
void eventPostBeginPlay();
};

// Class TAGame.PickupTimer_TA
// 0x0008 (0x02E0 - 0x02E8)
class APickupTimer_TA : public ACarComponent_TA
{
public:
int32_t                TimeTillItem;            // 0x02E0 (0x0004)
[0x0000008000002020] (CPF_Net | CPF_Transient)
int32_t                MaxTimeTillItem;         // 0x02E4 (0x0004)
[0x0000008000002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PickupTimer_TA");
}

return uClassPointer;
};

void SetMaxTimeTillItem(int32_t NewMaxTime);
void SetTimeTillItem(int32_t NewTime);
```

```
};

// Class TAGame.GFxHUD_GameEditor_TA
// 0x0010 (0x0540 - 0x0550)
class AGFxHUD_GameEditor_TA : public AGFxHUD_TA
{
public:
    class UGFxData_TrainingModeEditor_TA*      EditorData;           // 0x0540
    (0x0008) [0x0000008000002000] (CPF_Transient)
    class AGameEvent_TrainingEditor_TA*        TrainingGameEvent;    // 0x0548
    (0x0008) [0x0000008000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_GameEditor_TA");
        }

        return uClassPointer;
    }

    void UpdateCarData();
    void SetGameEvent(class AGameEvent_TA* InGameEvent);
    void InitGFx();
};

// Class TAGame.GFxHUD_Soccar_TA
// 0x0018 (0x0540 - 0x0558)
class AGFxHUD_Soccar_TA : public AGFxHUD_TA
{
public:
    class AGameEvent_Soccar_TA*                SoccarGame;          // 0x0540
    (0x0008) [0x0000008000002000] (CPF_Transient)
    class UGFxData_ServerPerformanceStats_TA*   GFxServerPerformanceStats; // 0x0548 (0x0008) [0x0000000000000000]
    class UPreMatchLobby_TA*                   PreMatchLobby;       // 0x0550
    (0x0008) [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_Soccar_TA");
        }

        return uClassPointer;
    }
}
```

```

class APRI_TA* GetSelectedPRI();
class APRI_TA* GetHoveredPRI();
void UpdateCarData();
void HandleReplicatedServerStats();
void HandleShowIntroScene();
void HandleTieBreakDecisionChanged();
void HandleMatchWinnerSet(class AGameEvent_Soccar_TA* G);
void HandlePlayerRemoved(class AGameEvent_TA* G, class APRI_TA* InPRI);
void HandlePlayerAdded(class AGameEvent_TA* G, class APRI_TA* InPRI);
void UpdateReplayData();
void HandleScoreDataChanged(class AReplayDirector_TA* ReplayDirector);
void HandleReplayDirectorSet(class AGameEvent_Soccar_TA* G);
void HandleNextMatchTimeUpdated(class AGameEvent_TA* InGameEvent);
void HandleWaitTimeUpdated();
void HandleOvertimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void HandleGameTimeUpdated(class AGameEvent_Soccar_TA* InGameEvent);
void OnAllTeamsCreated();
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void eventDestroyed();
void Init();
};


```

```

// Class TAGame.GFxHUD_KnockOut_TA
// 0x0038 (0x0558 - 0x0590)
class AGFxHUD_KnockOut_TA : public AGFxHUD_Soccar_TA
{
public:
    class AGameEvent_KnockOut_TA* KnockOutGame; // 0x0558
    (0x0008) [0x0001008000002000] (CPF_Transient)
    class UGFxData_CarKnockOut_TA* KnockOutCarData; // 0x0560
    (0x0008) [0x0001008000002000] (CPF_Transient)
    class UGameViewportClient_TA* ViewportClient; // 0x0568
    (0x0008) [0x0001000000002000] (CPF_Transient)
    int32_t DefaultMinZOffsetValue; // 0x0570 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t DefaultMaxZOffsetValue; // 0x0574 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t LowMinZOffsetClampValue; // 0x0578 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t HighMinZOffsetClampValue; // 0x057C (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t LowMaxZOffsetClampValue; // 0x0580 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t HighMaxZOffsetClampValue; // 0x0584 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t MinSqDistance; // 0x0588 (0x0004)
    [0x0001000000000001] (CPF_Edit)
    int32_t MaxSqDistance; // 0x058C (0x0004)
    [0x0001000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_KnockOut_TA");
}

return uClassPointer;
};

void __GFxHUD_KnockOut_TA__HandleMatchEnded_0x1(class APRI_TA* PRI);
void UpdateCarAttackTarget(unsigned long bPrevAttackTargetActive);
float CalculateKnockoutItemTargetZOffset();
void HandleMatchEnded(class AGameEvent_KnockOut_TA* InGameEvent);
void HandleKnockoutGameTimeUpdated(class AGameEvent_KnockOut_TA* InGameEvent);
void UpdateCarData();
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void eventDestroyed();
void Init();
};

// Class TAGame.GFxHUD_Replay_TA
// 0x0020 (0x05A0 - 0x05C0)
class AGFxHUD_Replay_TA : public AGFxHUD_Spectator_TA
{
public:
class AGameInfo_Replay_TA* Game; // 0x05A0 (0x0008)
[0x0000000000000000] (CPF_Transient)
float PrevKeyframeBufferTime; // 0x05A8 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bUserPaused : 1; // 0x05AC (0x0004)
[0x0000000000000000] [0x00000001] (CPF_Transient)
unsigned long bPausedForEndOfReplay : 1; // 0x05AC (0x0004)
[0x0000000000000000] [0x00000002] (CPF_Transient)
TArray<class UProductAsset_TA*> ReferencedProducts; // 0x05B0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_Replay_TA");
}

return uClassPointer;
};

void eventDestroyed();
void ReplayPause();
void ScrubTimeline(float DeltaTime);
void OpenSpeedMenu();

```

```

void PrevKeyframe();
void NextKeyframe();
void RemoveKeyframe();
void AddKeyframe();
void DrawLogs();
void DrawHUD();
void SetGameEvent(class AGameEvent_TA* InGameEvent);
void HandleGameEventSet(class AGameInfo_Replay_TA* InGame);
void SetPausedForEndOfReplay(unsigned long bPaused);
void HandlePostTimeSkip(class UReplay_TA* Replay);
void HandlePreTimeSkip(class UReplay_TA* Replay);
void HandlePlaybackStopped(class UReplay_TA* Replay);
void HandleReplayStarted(class AGameInfo_Replay_TA* InGame);
void InitGFx();
};

// Class TAGame.GFxHUD_Season_TA
// 0x0000 (0x0558 - 0x0558)
class AGFxHUD_Season_TA : public AGFxHUD_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_Season_TA");
}

return uClassPointer;
};

};

// Class TAGame.GFxHUD_Training_TA
// 0x0008 (0x0558 - 0x0560)
class AGFxHUD_Training_TA : public AGFxHUD_Soccar_TA
{
public:
class UGfxData_Training_TA*           TrainingData;          // 0x0558 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxHUD_Training_TA");
}
}

```

```

return uClassPointer;
};

void OpenShotSelectionMenu();
void SetGameEvent(class AGameEvent_TA* InGameEvent);
};

// Class TAGame.GFxModal_Ban_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Ban_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Ban_TA");
}

return uClassPointer;
};

class UGFxModal_Ban_TA* SetCitations(TArray<class FString> Citations);
};

// Class TAGame.GFxModal_FriendJoin_TA
// 0x0030 (0x00C8 - 0x00F8)
class UGFxModal_FriendJoin_TA : public UGFxModal_X
{
public:
class FString           ConnectingTitleString;          // 0x00C8 (0x0010)
[0x0000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           ConnectingBodyString;          // 0x00D8 (0x0010)
[0x0000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           JoiningGameFailedString;      // 0x00E8 (0x0010)
[0x0000000000408003] (CPF_Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_FriendJoin_TA");
}

return uClassPointer;
};

```

```

void HandlePasswordRequired();
void HandleGameInviteComplete(unsigned long bSuccess, class FString FailReason);
void SetGFxObject(class UGFxObject* Obj);
void Close();
class UOnlineGame_X* GetOnlineGame();
};

// Class TAGame.GFxModal_RankedReconnect_TA
// 0x0028 (0x00C8 - 0x00F0)
class UGFxModal_RankedReconnect_TA : public UGFxModal_X
{
public:
class FString           ConnectingTitleString;          // 0x00C8 (0x0010)
[0x0000000000408003] (CPF>Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class FString           ConnectingBodyString;          // 0x00D8 (0x0010)
[0x0000000000408003] (CPF>Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)
class UFindServerTask_X* Task;                         // 0x00E8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_RankedReconnect_TA");
}

return uClassPointer;
};

void HandleTaskError(class UError* Error);
void HandleJoin(unsigned long bSuccess, class FString FailReason);
void StartJoin(struct FServerReservationData Reservation);
void StartReconnect(struct FServerReservationData Reservation);
void SetGFxObject(class UGFxObject* Obj);
void Close();
class UOnlineGame_X* GetOnlineGame();
};

// Class TAGame.GFxModal_TextInput_TA
// 0x0018 (0x00C8 - 0x00E0)
class UGFxModal_TextInput_TA : public UGFxModal_X
{
public:
struct FScriptDelegate           Callback;             // 0x00C8 (0x0018)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_TextInput_TA");
}

return uClassPointer;
};

void HandleInputComplete(class FString InputText, unsigned long bCanceled);
void SetData(class UGFxObject* Data);
class UGFxModal_TextInput_TA* SetTextInputData(struct FScriptDelegate InCallback, unsigned
long bDisplayAsPassword, class FString DefaultText, int32_t MaxLength);
};

// Class TAGame.GFxModal_Tutorial_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Tutorial_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Tutorial_TA");
}

return uClassPointer;
};

void HandleButtonClicked(int32_t Index);
};

// Class TAGame.GFxModal_Warning_TA
// 0x0000 (0x00C8 - 0x00C8)
class UGFxModal_Warning_TA : public UGFxModal_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxModal_Warning_TA");
}

return uClassPointer;
}
```

```
};

};

// Class TAGame.NameplateRenderTarget_TA
// 0x0008 (0x01B8 - 0x01C0)
class UNameplateRenderTarget_TA : public UScriptedTexture
{
public:
class UGFxMoviePlayer_X*           NameplatesMovie;          // 0x01B8
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NameplateRenderTarget_TA");
}

return uClassPointer;
};

void Render(class UCanvas* C);
};

// Class TAGame.PsyNetService_CrewUnknownStatus_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_CrewUnknownStatus_TA : public UPsyNetClientService_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_CrewUnknownStatus_TA");
}

return uClassPointer;
};

};

// Class TAGame.RankedReconnectSave_TA
// 0x0078 (0x00C8 - 0x0140)
class URankedReconnectSave_TA : public USaveObject_TA
{
public:
```

```
struct FServerReservationData          Reservation;           // 0x00C8 (0x0070)
[0x0000004000400000] (CPF_NeedCtorLink)
uint64_t                         EpochTime;           // 0x0138 (0x0008)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RankedReconnectSave_TA");
}

return uClassPointer;
};

class USaveObject_TA* Reconcile(class USaveObject_TA* Remote);
void ClearRankedReconnect();
bool RankedReconnectAvailable();
void SetRankedReconnect(int32_t InPlaylistID, struct FActiveServerData& Server);
};

// Class TAGame.GFxStayAsPartyDisplayButton_TA
// 0x0000 (0x0060 - 0x0060)
class UGFxStayAsPartyDisplayButton_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GFxStayAsPartyDisplayButton_TA");
}

return uClassPointer;
};

};

// Class TAGame.GoalVolume_Hoops_TA
// 0x0000 (0x02B0 - 0x02B0)
class AGoalVolume_Hoops_TA : public AGoalVolume_TA
{
public:

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.GoalVolume_Hoops_TA");
}

return uClassPointer;
};

};

// Class TAGame.HauntedBallTrapTrigger_TA
// 0x0068 (0x0270 - 0x02D8)
class AHauntedBallTrapTrigger_TA : public ADynamicMeshActor_TA
{
public:
class ALensFlareSource*           LensFlare;           // 0x0270 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t                           TeamIndex;          // 0x0278 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AGoalVolume_TA*            GoalVolume;         // 0x0280 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                            TrapResetTime;      // 0x0288 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                            MinLensFlareColorValue; // 0x028C (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FName                      MICParamName;       // 0x0290 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant*   MIC_TrapVolumeMat; // 0x0298
(0x0008) [0x0000000000002000] (CPF_Transient)
class UMaterialInstanceConstant*   MIC_TrapOutlineMat; // 0x02A0
(0x0008) [0x0000000000002000] (CPF_Transient)
unsigned long                     bIsActive : 1;        // 0x02A8 (0x0004)
[0x0000000000000000] [0x00000001]
class ABall_Haunted_TA*          HauntedBall;        // 0x02B0 (0x0008)
[0x0000000000000000]
class AGameEvent_Soccar_TA*      SoccarEvent;        // 0x02B8 (0x0008)
[0x0000000000000000]
class UParticleSystemComponent*   BeamPSC;           // 0x02C0
(0x0008) [0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class UParticleSystem*           BeamPSArchetype;    // 0x02C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName                      BeamPSPParam;       // 0x02D0 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.HauntedBallTrapTrigger_TA");
}

return uClassPointer;
};

void SetTimeInGoalValues(float Value);
void ResetTrap();
void SetTrapVisuals(unsigned long bInIsActive);
void SetTrapActivity(unsigned long bInIsActive);
bool GetIsTrapActive();
void HandleIsTrappedChanged(class ABall_Haunted_TA* Ball, unsigned long bInIsTrapped, class UGoal_TA* Goal);
void HandleLastTeamTouchChanged(class ABall_Haunted_TA* Ball, uint8_t InLastTeamTouch);
void HandleDeactivatedGoalIndexChanged();
void HandleBallAdded(class AGameEvent_Soccar_TA* GameEvent, class ABall_Haunted_TA* Ball);
void HandleGameEventRemoved(class AGameEvent_TA* GameEvent);
void HandleGameEventAdded(class AGameEvent_Soccar_TA* GameEvent);
void eventTick(float DeltaTime);
void eventDestroyed();
void eventPostBeginPlay();
};

// Class TAGame.HUDActor_TA
// 0x0020 (0x0268 - 0x0288)
class AHUDActor_TA : public AActor
{
public:
class AHUD*           HUD;           // 0x0268 (0x0008)
[0x0000000000000000]
struct FScriptDelegate _RenderDelegate__Delegate; // 0x0270
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.HUDActor_TA");
}

return uClassPointer;
};

static class AHUDActor_TA* Create(class AHUD* InHUD, struct FScriptDelegate InRenderDelegate);
void eventDestroyed();
void Init(class AHUD* InHUD);
void eventPostRenderFor(class APlayerController* PC, class UCanvas* Canvas, struct FVector CameraPosition, struct FVector CameraDir);
void RenderDelegate(class UCanvas* Canvas);

```

```
};

// Class TAGame.IDefaultProduct
// 0x0000 (0x0060 - 0x0060)
class UDefaultProduct : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IDefaultProduct");
}

return uClassPointer;
};

int32_t GetDefaultProductID();
};

// Class TAGame.IdleAction_TA
// 0x0000 (0x0060 - 0x0060)
class UIdleAction_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdleAction_TA");
}

return uClassPointer;
};

};

// Class TAGame.IdleFeature_TA
// 0x0008 (0x0060 - 0x0068)
class UIdleFeature_TA : public UObject
{
public:
float IdleActionTime; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
float IdleWarningTime; // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)
```

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.IdleFeature_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.IdleBanFeature_TA  
// 0x0000 (0x0068 - 0x0068)  
class UIdleBanFeature_TA : public UIdleFeature_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.IdleBanFeature_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.IdleSystem_TA  
// 0x0000 (0x0060 - 0x0060)  
class UIdleSystem_TA : public UObject  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.IdleSystem_TA");  
}  
  
return uClassPointer;
```

```
};

static bool __IdleSystem_TA__UpdateIdle_0x1(class UChildConnection* ChildCon);
static float GetLastActiveTime(class UNetConnection* Connection);
static void UpdateIdle(class UIdleTimer_TA* Timer, class AWorldInfo* WorldInfo, class
UIdleFeature_TA* Feature, class UIdlePlayer_TA* IdlePlayer, class UNetConnection* Connection,
class APlayerController_TA* PC, class APRI_TA* PRI);
static void InitIdleStartTime(class AWorldInfo* WorldInfo, class UActiveGameState_TA*
GameState, class UIdlePlayer_TA* IdlePlayer);
static void AddIdleConnection(class UIdleFeature_TA* Feature, class UNetConnection*
Connection);
static void StopIdleTimer(class UIdleFeature_TA* Feature, class UActiveGameState_TA*
GameState);
static void StartIdleTimer(class UIdleFeature_TA* Feature, class UActiveGameState_TA*
GameState);
};

// Class TAGame.IdleTimer_TA
// 0x0000 (0x0060 - 0x0060)
class UIdleTimer_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdleTimer_TA");
}

return uClassPointer;
};

};

// Class TAGame.IdlePlayer_TA
// 0x0004 (0x0060 - 0x0064)
class UIdlePlayer_TA : public UObject
{
public:
float StartTime; // 0x0060 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdlePlayer_TA");
}
}
```

```
}

return uClassPointer;
};

};

// Class TAGame.IdleWarning_TA
// 0x0000 (0x0268 - 0x0268)
class AldleWarning_TA : public AOwnerReplicatedActor_ORS
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdleWarning_TA");
}

return uClassPointer;
};

};

// Class TAGame.IdleBanSystem_TA
// 0x0000 (0x0060 - 0x0060)
class UidleBanSystem_TA : public UidleSystem_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdleBanSystem_TA");
}

return uClassPointer;
};

static void PerformIdleAction(class UidleBanFeature_TA* Feature, class UidleAction_TA* Action,
class UidlePlayer_TA* IdlePlayer, class UNetConnection* Connection, class
APlayerController_TA* PC);
};

// Class TAGame.TourGameUpdate_TA
// 0x0024 (0x0060 - 0x0084)
```

```

class UTourGameUpdate_TA : public UObject
{
public:
TArray<struct FUniqueNetId>           JoinedPlayers;          // 0x0060 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId>           IdlePlayers;          // 0x0070 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
unsigned long             bHasNetworkIssues : 1;        // 0x0080 (0x0004)
[0x0001004000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourGameUpdate_TA");
}

return uClassPointer;
};

void __TourGameUpdate_TA__Construct_0x1(class AGameEvent_TA* GameEvent);
void __TourGameUpdate_TA__Construct_0x2(class AGameEvent_TA* G, class APRI_TA* PRI);
bool __TourGameUpdate_TA__GetNoShowPlayers_0x1(struct FUniqueNetId Id);
struct FUniqueNetId __TourGameUpdate_TA__GetLeavers_0x2(struct FReservationData P);
bool __TourGameUpdate_TA__GetLeavers_0x1(struct FReservationData P);
struct FTourPlayerStats __TourGameUpdate_TA__GetPlayersStats_0x2(class APRI_TA* PRI);
class APRI_TA* __TourGameUpdate_TA__GetPlayersStats_0x1(class AController* P);
class URPC_X* SendUpdateGameRPC(class UTourGameUpdateDispatcher_TA* Dispatcher,
int32_t InGameNum, struct FTourMatchGame InGameData, uint64_t SeriesWinnerID);
TArray<struct FTourPlayerStats> GetPlayersStats();
void SetHasNetworkIssues(unsigned long bValue);
TArray<struct FUniqueNetId> GetLeavers();
TArray<struct FUniqueNetId> GetNoShowPlayers(class UTourGameUpdateDispatcher_TA*
Dispatcher);
void eventConstruct();
};

// Class TAGame.IdleKickFeature_TA
// 0x0000 (0x0068 - 0x0068)
class UIdeleKickFeature_TA : public UIdeleFeature_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdleKickFeature_TA");
}

```

```

}

return uClassPointer;
};

};

// Class TAGame.IdleKickSystem_TA
// 0x0000 (0x0060 - 0x0060)
class UIdleKickSystem_TA : public UIdleSystem_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IdleKickSystem_TA");
}

return uClassPointer;
};

static void HideIdleWarning(class UGFxData_LocalPlayer_TA* GFxData, class AIdleWarning_TA* Warning);
static void ShowIdleWarning(class UGFxData_LocalPlayer_TA* GFxData, class AIdleWarning_TA* Warning);
static void PerformIdleAction(class UIdleKickFeature_TA* Feature, class UIdleAction_TA* Action,
class UIdlePlayer_TA* IdlePlayer, class UNetConnection* Connection, class APlayerController_TA* PC);
};

// Class TAGame.IForcedProduct
// 0x0000 (0x0060 - 0x0060)
class UIForcedProduct : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IForcedProduct");
}

return uClassPointer;
};

```

```
int32_t GetForcedProductID();
};

// Class TAGame.IMatchmakingViewFilter_TA
// 0x0000 (0x0060 - 0x0060)
class UIMatchmakingViewFilter_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.IMatchmakingViewFilter_TA");
}

return uClassPointer;
};

TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,
TArray<struct FName>& SelectedPlaylists);
};

// Class TAGame.IMicroEventConfig_TA
// 0x0000 (0x0060 - 0x0060)
class UIMicroEventConfig_TA : public UInterface
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame. IMicroEventConfig_TA");
}

return uClassPointer;
};

bool IsActiveMicroEventPlaylist(int32_t PlaylistId);
class UMicroEvent_TA* GetNextActiveMicroEvent();
};

// Class TAGame.MusicTracklist_TA
// 0x0010 (0x0060 - 0x0070)
class UMusicTracklist_TA : public UObject
{
public:
```

```

TArray<class UMusicTrack_TA*> Tracks; // 0x0060 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTracklist_TA");
}

return uClassPointer;
};

class UMusicTracklist_TA* AddLastTrackAction(struct FScriptDelegate Action);
class UIMusicTracklist_TA* Rebuild(class UMusicTracklistFactory_TA* Builder);
int32_t Num();
void NextTrack();
};

// Class TAGame.InMapScoreboard_TA
// 0x00AC (0x0268 - 0x0314)
class AInMapScoreboard_TA : public AActor
{
public:
class AStaticMeshActor* StaticMesh; // 0x0268 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UMaterialInstanceConstant* MaterialInstance; // 0x0270
(0x0008) [0x0000000000000001] (CPF_Edit)
int32_t MaterialIndex; // 0x0278 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t TeamIndex; // 0x027C (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat GoalAnimation; // 0x0280 (0x0018)
[0x00000000040001] (CPF_Edit | CPF_NeedCtorLink)
float GoDuration; // 0x0298 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FName GoSwitch; // 0x029C (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName GoalXPos; // 0x02A4 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName CountdownSwitch; // 0x02AC (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName CountdownToggle; // 0x02B4 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName ScoreToggle; // 0x02BC (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName ScoreParam; // 0x02C4 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName CountdownValueParam; // 0x02CC (0x0008)
[0x0000000000000001] (CPF_Edit)
TArray<struct FScoreboardMaterialOverride> ExtraScoreboardMICOverrides; //

```

```

0x02D8 (0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
int32_t LastCountdownTime; // 0x02E8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float ReplayCountdownTime; // 0x02EC (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long bHasReplayCountdownTime : 1; // 0x02F0
(0x0004) [0x0000000000002000] [0x00000001] (CPF_Transient)
class AGameEvent_Soccar_TA* GameEvent; // 0x02F8 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FName LastState; // 0x0300 (0x0008)
[0x0000000000002000] (CPF_Transient)
class UMaterialInstanceConstant* MIC; // 0x0308 (0x0008)
[0x0000000000002000] (CPF_Transient)
float GoalTime; // 0x0310 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InMapScoreboard_TA");
}

return uClassPointer;
};

void HandleReplayGoalScored(class AGameEvent_Soccar_TA* InEvent, uint8_t ScoredOnTeam);
void UpdateExtraMicOverrides(float ValueToUse);
void HideGo();
void ShowGo();
void HideCountdown();
void ShowCountdown();
void HideScore();
void ShowScore();
void UpdateScore(unsigned long bNoReplay);
void DisableAllSwitches();
void HandleCountdownTimeUpdated(class AGameEvent_TA* InEvent, int32_t Seconds);
void HandleGameStateChanged(class AGameEvent_TA* InEvent);
void HandleScoreUpdated(class ATeam_TA* Team);
void HandleTeams(class AGameEvent_Team_TA* G);
void SetUpEvents();
void HandleGameEvent(class AGameEvent_Soccar_TA* SoccarGame);
class UMaterialInstanceConstant* CreateAndAddMicToMesh(class UMaterialInstanceConstant* ParentMICInstance, int32_t MaterialIndexToUse);
void eventPostBeginPlay();
};

// Class TAGame.InputBufferUtil_TA
// 0x0000 (0x0060 - 0x0060)
class UInputBufferUtil_TA : public UObject
{

```

```

public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.InputBufferUtil_TA");
}

return uClassPointer;
};

static TArray<struct FClientFrameData> MergeFrames(TArray<struct FClientFrameData>& ArrayA, TArray<struct FClientFrameData>& ArrayB);
};

// Class TAGame.ItemPool_TA
// 0x0010 (0x0060 - 0x0070)
class UItemPool_TA : public UObject
{
public:
TArray<struct FRandomWeight> Items; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemPool_TA");
}

return uClassPointer;
};

void OnCarDestroyedWithItem(class ACar_TA* Car);
void GiveItem(class ACar_TA* Car);
bool PreviewItem(class ACar_TA* Car);
bool GiveItemForRespawn(class ACar_TA* Car);
void OnGoalScored();
void OnNewRound();
void OnInit();
int32_t GetConcurrentItemCount();
void Init();
};

// Class TAGame.ItemPoolCycle_TA
// 0x0020 (0x0070 - 0x0090)
class UItemPoolCycle_TA : public UItemPool_TA

```

```

{
public:
TArray<class ASpecialPickup_TA*> RemainingItems; // 0x0070
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class ASpecialPickup_TA* LastDispensedItem; // 0x0080 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ASpecialPickup_TA* DemolishedItem; // 0x0088 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemPoolCycle_TA");
}

return uClassPointer;
};

void OnCarDestroyedWithItem(class ACar_TA* Car);
void RefillPool();
void ApplyItemToCar(class ACar_TA* Car, class ASpecialPickup_TA* Item);
void GiveItem(class ACar_TA* Car);
bool GiveItemForRespawn(class ACar_TA* Car);
void OnNewRound();
void OnInit();
};

// Class TAGame.ItemPoolSelection_TA
// 0x0038 (0x0070 - 0x00A8)
class UItemPoolSelection_TA : public UItemPool_TA
{
public:
int32_t ConcurrentItemCount; // 0x0070 (0x0004)
[0x0001000000000001] (CPF_Edit)
unsigned long bPreviewItems : 1; // 0x0074 (0x0004)
[0x0001000000000001] [0x00000001] (CPF_Edit)
TArray<class ASpecialPickup_TA*> LastDispensedItems; // 0x0078
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class ASpecialPickup_TA*> DemolishedItems; // 0x0088
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t OriginalPickupCooldownSeconds; // 0x0098 (0x0004)
[0x0001000000002000] (CPF_Transient)
int32_t CooldownSecondsAtPreviousGoal; // 0x009C (0x0004)
[0x0001000000002000] (CPF_Transient)
class ASpecialPickup_TA* ActivePickup; // 0x00A0 (0x0008)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemPoolSelection_TA");
}

return uClassPointer;
};

void OnCarDestroyedWithItem(class ACar_TA* Car);
void ApplyItemsToCar(class ACar_TA* Car, TArray<class ASpecialPickup_TA*>& ApplyItems);
void HandleItemDeactivated(class ACar_TA* Car, class ASpecialPickup_TA* Item);
void HandleItemActivated(class ACar_TA* Car, class ASpecialPickup_TA* Item);
void AddItemToSlot(class ASpecialPickup_TA* Item, int32_t SlotIndex, class ACar_TA* Car);
void RollItems(class ACar_TA* Car);
void GiveItem(class ACar_TA* Car);
bool PreviewItem(class ACar_TA* Car);
bool GiveItemForRespawn(class ACar_TA* Car);
int32_t GetConcurrentItemCount();
void OnGoalScored();
void OnNewRound();
void OnInit();
};

// Class TAGame.ItemPoolWeightMultiplier_TA
// 0x0018 (0x0070 - 0x0088)
class UItemPoolWeightMultiplier_TA : public UItemPool_TA
{
public:
int32_t PickWeightMultiplier; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bResetWeightsOnGoal : 1; // 0x0074 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class ASpecialPickup_TA* LastDispensedItem; // 0x0078 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ASpecialPickup_TA* DemolishedItem; // 0x0080 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ItemPoolWeightMultiplier_TA");
}

return uClassPointer;
};

void OnCarDestroyedWithItem(class ACar_TA* Car);
void ApplyItemToCar(class ACar_TA* Car, class ASpecialPickup_TA* Item);

```

```

void GiveItem(class ACar_TA* Car);
bool GiveItemForRespawn(class ACar_TA* Car);
void ResetItemWeights();
void OnNewRound();
void OnInit();
};

// Class TAGame.JoinGameConnectionChecker_TA
// 0x0030 (0x0060 - 0x0090)
class UJoinGameConnectionChecker_TA : public UObject
{
public:
    class UConnectionQualityConfig_TA* ConnectionQualityConfig; // 0x0060
    (0x0008) [0x0000800000002000] (CPF_Transient)
    class AGameEvent_TA* GameEvent; // 0x0068 (0x0008)
    [0x0000800000002000] (CPF_Transient)
    class UOnlineGame_X* OnlineGame; // 0x0070 (0x0008)
    [0x0000800000002000] (CPF_Transient)
    struct FScriptDelegate __ShowConnectionError__Delegate; // 0x0078
    (0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.JoinGameConnectionChecker_TA");
        }

        return uClassPointer;
    };

    bool __JoinGameConnectionChecker_TA__Construct_0x1(class UPlayer* P);
    void ShowConnectionErrorIfWaiting();
    bool IsPlayerWaiting();
    bool IsInMainMenu();
    void HandleDisconnected(class APlayerController* PC);
    void HandleJoinGameComplete(unsigned long bSuccess, class FString FailReason);
    void Remove(class ULocalPlayer_TA* LocalPlayer);
    void eventConstruct();
    void ShowConnectionError();
};

// Class TAGame.K3SUserPermissionsError_TA
// 0x0020 (0x0060 - 0x0080)
class UK3SUserPermissionsError_TA : public UObject
{
public:
    class FString ErrorCode; // 0x0060 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class FString Message; // 0x0070 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
};

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.K3SUserPermissionsError_TA");
}

return uClassPointer;
};

};

// Class TAGame.LightBarComponent_TA
// 0x002F (0x009D - 0x00CC)
class ULightBarComponent_TA : public UActorComponent
{
public:
class ATeam_TA* Team; // 0x00A0 (0x0008)
[0x0000004000002000] (CPF_Transient)
unsigned long bStrobeActive : 1; // 0x00A8 (0x0004)
[0x0000004000002000] [0x00000001] (CPF_Transient)
unsigned long bRandStrobeColor : 1; // 0x00A8 (0x0004)
[0x0000004000002000] [0x00000002] (CPF_Transient)
int32_t StrobeCount; // 0x00AC (0x0004)
[0x0000004000002000] (CPF_Transient)
int32_t MaxStrobeCount; // 0x00B0 (0x0004)
[0x0000004000002000] (CPF_Transient)
float StrobeTime; // 0x00B4 (0x0004)
[0x0000004000002000] (CPF_Transient)
TArray<struct FColor> StrobeColorList; // 0x00B8 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
int32_t StrobeColorIndex; // 0x00C8 (0x0004)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LightBarComponent_TA");
}

return uClassPointer;
};

void HandleBoostAmountChanged(class ACarComponent_Boost_TA* Boost);
void HandleTeamChanged(class APRI_X* PRI);
void Strobe_MVP(class ATeam_TA* InTeam);

```

```

void Strobe_GoalScored(int32_t ScoredOnTeam);
void StrobeColor();
void StartStrobeColor(float Duration, float InStrobeCount, unsigned long bRandColor);
void ResetColor();
void SetColor(struct FColor NewColor);
};

// Class TAGame.LightEnvironment_TA
// 0x0004 (0x018C - 0x0190)
class ULightEnvironment_TA : public UParticleLightEnvironmentComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LightEnvironment_TA");
}

return uClassPointer;
};

};

// Class TAGame.LoadoutRenderer_TA
// 0x0030 (0x0060 - 0x0090)
class ULoadoutRenderer_TA : public UObject
{
public:
class UThumbnailRenderer_TA*           ThumbnailRenderer;          // 0x0060
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductLoader_TA*              Loader;                  // 0x0068 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)
class ATeam_TA*                      TeamArchetype;          // 0x0070 (0x0008)
[0x0000000000002000] (CPF_Transient)
struct FScriptDelegate                _EventLoadoutRendered__Delegate; // 0x0078
(0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutRenderer_TA");
}

return uClassPointer;
};

```

```
void HandleAllProductsLoaded(class UProductLoader_TA* InLoader);
void RenderLoadout(struct FClientLoadoutData Loadout, class ATeam_TA* InTeamArchetype,
struct FScriptDelegate Callback);
void EventLoadoutRendered(class ULoadoutRenderer_TA* Renderer, class UTexture2D* Tex);
};

// Class TAGame.LoadoutUtils_TA
// 0x0000 (0x0060 - 0x0060)
class ULoadoutUtils_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutUtils_TA");
}

return uClassPointer;
};

static void ConformRandomizedLoadoutData(class ULoadout_TA* Loadout);
static void FixupDeprecatedTeamPaint(class ULoadout_TA* Loadout);
static void ConvertFromOldCustomPaint(class ULoadout_TA* Loadout, class UCarColorSet_TA*
OldSet, class UCarColorSet_TA* NewSet);
static void ConvertFromOldTeamPaint(class ULoadout_TA* Loadout, class UCarColorSet_TA*
OldSet, class UCarColorSet_TA* NewSet);
static void ConvertFromOldLoadout(class ULoadout_TA* OldLoadout, class ULoadout_TA*
NewLoadout, int32_t TeamIndex);
};

// Class TAGame.LocalizationConfig_TA
// 0x0000 (0x0088 - 0x0088)
class ULocalizationConfig_TA : public ULocalizationConfig_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LocalizationConfig_TA");
}

return uClassPointer;
};
```

```
void Apply();
};

// Class TAGame.LocalPlayerSkillSync_TA
// 0x0018 (0x0060 - 0x0078)
class ULocalPlayerSkillSync_TA : public UObject
{
public:
class UOnlineGame_X*           OnlineGame;          // 0x0060 (0x0008)
[0x0000800000000000]
class UOnlineGameSkill_X*       OnlineSkill;        // 0x0068 (0x0008)
[0x0000800000000000]
class UPsyNet_X*               PsyNet;            // 0x0070 (0x0008)
[0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LocalPlayerSkillSync_TA");
}

return uClassPointer;
};

void __LocalPlayerSkillSync_TA__Construct_0x1(class UOnlinePlayerAuthentication_X* Auth);
void SyncOnlinePlayerSkill(class UOnlinePlayer_X* Player);
void SyncSkillsDelayed(float Delay);
void SyncSkills();
void HandlePsyNetConnection();
void eventConstruct();
};

// Class TAGame.MapGroups_TA
// 0x0000 (0x0060 - 0x0060)
class UMapGroups_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapGroups_TA");
}

return uClassPointer;
};
```

```

};

static TArray<class UMapData_TA*> __MapGroups_TA__BuildMaps_0x1(struct FMapGroup Group);
static TArray<class UMapData_TA*> BuildMaps(TArray<struct FMapGroup>& Groups);
static TArray<struct FMapGroup> BuildGroups(TArray<class UMapData_TA*>& AvailableMaps,
TArray<struct FMapGroup>& SourceGroups);
};

// Class TAGame.MapPrefsMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UMapPrefsMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MapPrefsMetrics_TA");
}

return uClassPointer;
};

void MapPrefs(TArray<struct FMapPrefsMetric> Playlists);
struct FMapPrefsMetric GetMetric(struct FPlaylistMapPrefs& InPrefs);
TArray<struct FMapPrefsMetric> GetAllMetrics(TArray<struct FPlaylistMapPrefs>& InMapPrefs);
void RecordMapPrefs(TArray<struct FPlaylistMapPrefs>& InMapPrefs);
};

// Class TAGame.MatchCompleteMetrics_TA
// 0x0010 (0x0080 - 0x0090)
class UMatchCompleteMetrics_TA : public UMetricsGroup_X
{
public:
TArray<struct FMatchCompleteHistory> History; // 0x0080 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchCompleteMetrics_TA");
}

return uClassPointer;
};

```

```

void RewardsReceived(class FString MatchGuid, struct FUniqueNetId PlayerID, float
LatencyFromMatchComplete);
void RecordRewardsReceived(class FString MatchGuid, struct FUniqueNetId PlayerID);
void RecordMatchComplete(class FString MatchGuid, struct FUniqueNetId PlayerID);
int32_t GetHistoryIndex(class FString MatchGuid);
};

// Class TAGame.MatchmakingViewFilter_Casual_TA
// 0x0020 (0x0060 - 0x0080)
class UMatchmakingViewFilter_Casual_TA : public UObject
{
public:
class UIMicroEventConfig_TA*
(0x0008) [0x0000800000000000] MicroEventConfig_Object; // 0x0060
class UIMicroEventConfig_TA*
(0x0008) [0x0000800000000000] MicroEventConfig_Interface; // 0x0068
class UIOnlineGamePlaylists_X*
[0x0000800000000000] Playlists_Object; // 0x0070 (0x0008)
class UIOnlineGamePlaylists_X*
(0x0008) [0x0000800000000000] Playlists_Interface; // 0x0078

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilter_Casual_TA");
}

return uClassPointer;
};

bool __MatchmakingViewFilter_Casual_TA__GetAccessiblePlaylists_0x1(struct FName P);
TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,
TArray<struct FName>& SelectedPlaylists);
};

// Class TAGame.MatchmakingViewFilter_DebugMicroEvent_TA
// 0x0018 (0x0060 - 0x0078)
class UMatchmakingViewFilter_DebugMicroEvent_TA : public UObject
{
public:
class USpecialEventConfig_TA*
(0x0008) [0x0000800000000000] SpecialEventConfig; // 0x0060
class UIOnlineGamePlaylists_X*
[0x0000800000000000] Playlists_Object; // 0x0068 (0x0008)
class UIOnlineGamePlaylists_X*
(0x0008) [0x0000800000000000] Playlists_Interface; // 0x0070

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.MatchmakingViewFilter_DebugMicroEvent_TA");
}

return uClassPointer;
};

bool __MatchmakingViewFilter_DebugMicroEvent_TA__GetAccessiblePlaylists_0x1(struct
FName P);
TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,
TArray<struct FName>& SelectedPlaylists);
};

// Class TAGame.MatchmakingViewFilter_Unranked_TA
// 0x0020 (0x0060 - 0x0080)
class UMatchmakingViewFilter_Unranked_TA : public UObject
{
public:
class UIMicroEventConfig_TA*           MicroEventConfig_Object;      // 0x0060
(0x0008) [0x0000800000000000]
class UIMicroEventConfig_TA*           MicroEventConfig_Interface;   // 0x0068
(0x0008) [0x0000800000000000]
class UIOnlineGamePlaylists_X*         Playlists_Object;           // 0x0070 (0x0008)
[0x0000800000000000]
class UIOnlineGamePlaylists_X*         Playlists_Interface;        // 0x0078
(0x0008) [0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilter_Unranked_TA");
}

return uClassPointer;
};

bool __MatchmakingViewFilter_Unranked_TA__GetAccessiblePlaylists_0x1(struct FName P);
TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,
TArray<struct FName>& SelectedPlaylists);
};

// Class TAGame.MatchmakingViewFilter_MicroEvent_TA
// 0x0000 (0x0080 - 0x0080)
class UMatchmakingViewFilter_MicroEvent_TA : public UMatchmakingViewFilter_Unranked_TA
{

```

```

public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilter_MicroEvent_TA");
}

return uClassPointer;
};

TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,
TArray<struct FName>& SelectedPlaylists);
};

// Class TAGame.MatchmakingViewFilter_Ranked_TA
// 0x0010 (0x0060 - 0x0070)
class UMatchmakingViewFilter_Ranked_TA : public UObject
{
public:
class UIOnlineGamePlaylists_X* [0x0000800000000000] Playlists_Object; // 0x0060 (0x0008)
class UIOnlineGamePlaylists_X* (0x0008) [0x0000800000000000] Playlists_Interface; // 0x0068

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilter_Ranked_TA");
}

return uClassPointer;
};

TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,
TArray<struct FName>& SelectedPlaylists);
};

// Class TAGame.MatchmakingViewFilter_RankedSports_TA
// 0x0010 (0x0060 - 0x0070)
class UMatchmakingViewFilter_RankedSports_TA : public UObject
{
public:
class UIOnlineGamePlaylists_X* [0x0000800000000000] Playlists_Object; // 0x0060 (0x0008)
class UIOnlineGamePlaylists_X* Playlists_Interface; // 0x0068

```

(0x0008) [0x0000800000000000]

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.MatchmakingViewFilter_RankedSports_TA");  
}  
  
return uClassPointer;  
};  
  
TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,  
TArray<struct FName>& SelectedPlaylists);  
};  
  
// Class TAGame.MatchmakingViewFilter_UnrankedSports_TA  
// 0x0010 (0x0060 - 0x0070)  
class UMatchmakingViewFilter_UnrankedSports_TA : public UObject  
{  
public:  
class UIOnlineGamePlaylists_X* [0x0000800000000000] Playlists_Object; // 0x0060 (0x0008)  
class UIOnlineGamePlaylists_X* (0x0008) [0x0000800000000000] Playlists_Interface; // 0x0068  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class  
TAGame.MatchmakingViewFilter_UnrankedSports_TA");  
}  
  
return uClassPointer;  
};  
  
TArray<struct FName> GetAccessiblePlaylists(struct FName LastVisitedPlaylistFolderID,  
TArray<struct FName>& SelectedPlaylists);  
};  
  
// Class TAGame.MatchType_AdHoc_TA  
// 0x0000 (0x00A8 - 0x00A8)  
class UMatchType_AdHoc_TA : public UMatchType_Private_TA  
{  
public:  
  
public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_AdHoc_TA");
}

return uClassPointer;
};

struct FName GetMatchTypeName();
};

// Class TAGame.TourServerConfig_TA
// 0x0008 (0x0078 - 0x0080)
class UTourServerConfig_TA : public UOnlineConfig_X
{
public:
unsigned long           bBotTeamsEnabled : 1;          // 0x0078 (0x0004)
[0x0001000000000001] [0x00000001] (CPF_Edit)
float                  MidSeriesConnectionTimeout;    // 0x007C (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourServerConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.MatchType_PhysicsLogPlayback_TA
// 0x0000 (0x00A8 - 0x00A8)
class UMatchType_PhysicsLogPlayback_TA : public UMatchType_Tutorial_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MatchType_PhysicsLogPlayback_TA");
}
```

```

}

return uClassPointer;
};

bool AutoSpectate(class AController* NewPlayer);
bool ShouldAutoSelectTeam(class AController* NewPlayer);
};

// Class TAGame.MaxTimeComponent_TA
// 0x0008 (0x0060 - 0x0068)
class UMaxTimeComponent_TA : public UObject
{
public:
int32_t WarningTimeSeconds; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t TimeLimitSeconds; // 0x0064 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MaxTimeComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.MaxTimeEndGame_TA
// 0x0000 (0x0060 - 0x0060)
class UMaxTimeEndGame_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MaxTimeEndGame_TA");
}

return uClassPointer;
};

static int32_t __MaxTimeEndGame_TA__GetWinningTeam_0x1(class ATeam_TA* T);

```

```

static int32_t __MaxTimeEndGame_TA__GetWinningTeam_0x2(int32_t Sum, class APRI_TA* PRI);
static class ATeam_TA* GetWinningTeam(class AGameEvent_Soccar_TA* GameEvent, uint8_t& TieBreakDecision);
};

// Class TAGame.MaxTimeState_TA
// 0x0000 (0x0060 - 0x0060)
class UMaxTimeState_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MaxTimeState_TA");
}

return uClassPointer;
};

};

// Class TAGame.MaxTimeSystem_TA
// 0x0000 (0x0060 - 0x0060)
class UMaxTimeSystem_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MaxTimeSystem_TA");
}

return uClassPointer;
};

static void RemovedMaxTimeWarningData(class UGFxData_TourEvent_TA* GFxData_TourEvent,
class AMaxTimeWarningData_TA* _);
static void UpdateMaxTimeWarningData(class UGFxData_TourEvent_TA* GFxData_TourEvent,
class AMaxTimeWarningData_TA* MaxTimeWarningData);
static void HandleWarningMessage(class AGameEvent_Soccar_TA* GameEvent, class
UMaxTimeWarning_TA* MaxTimeWarningData);
static void ReceivedMaxTimeEndGame(class UMaxTimeComponent_TA* MaxTimeComponent,
class AGameEvent_Soccar_TA* GameEvent, class UMaxTimeEndGame_TA* MaxTimeEndGame);
static void ClearMaxTimeSystem(class UMaxTimeComponent_TA* MaxTimeComponent, class

```

```

AGameEvent_Soccar_TA* GameEvent);
static void OnGameFinished(class UMaxTimeComponent_TA* MaxTimeComponent, class
AGameEvent_Soccar_TA* GameEvent, class UGameEventFinishedState_TA* _);
static void StartMaxTimer(class UMaxTimeComponent_TA* MaxTimeComponent, class
UGameEventMatchActive_TA* GameEventMatchActive, class AGameEvent_Soccar_TA*
GameEvent);
};

// Class TAGame.MaxTimeWarningData_TA
// 0x0010 (0x0270 - 0x0280)
class AMaxTimeWarningData_TA : public AReplicatedActor_ORS
{
public:
    uint64_t EndGameEpochTime; // 0x0270 (0x0008)
    [0x0000000000000020] (CPF_Net)
    uint64_t EndGameWarningEpochTime; // 0x0278 (0x0008)
    [0x0000000000000020] (CPF_Net)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.MaxTimeWarningData_TA");
        }

        return uClassPointer;
    };
};

// Class TAGame.MaxTimeWarning_TA
// 0x0000 (0x0060 - 0x0060)
class UMaxTimeWarning_TA : public UObject
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.MaxTimeWarning_TA");
        }

        return uClassPointer;
    };
};

```

```
// Class TAGame.MenuSequence_MusicStingers_TA
// 0x0000 (0x00A8 - 0x00A8)
class UMenuSequence_MusicStingers_TA : public UMenuSequence_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuSequence_MusicStingers_TA");
}

return uClassPointer;
};

void LeaveSequence(class UMenuSequence_TA* NextSequence);
void EnterSequence(class UMenuSequence_TA* PrevSequence);
};

// Class TAGame.SeqEvent_PreviewMusicStingerVisualizer_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_PreviewMusicStingerVisualizer_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.SeqEvent_PreviewMusicStingerVisualizer_TA");
}

return uClassPointer;
};

};

// Class TAGame.MenuSequence_PremiumGarage_TA
// 0x000C (0x00A8 - 0x00B4)
class UMenuSequence_PremiumGarage_TA : public UMenuSequence_TA
{
public:
float FadeTime; // 0x00A8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float FadeOutDelay; // 0x00AC (0x0004)
[0x0000000000000001] (CPF_Edit)
```

```
float TurnTableYawOffset; // 0x00B0 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuSequence_PremiumGarage_TA");
}

return uClassPointer;
};

class UShowroomMetrics_TA* GetShowroomMetrics();
void LeaveFadeOut();
void HandleLeaveComplete();
void EnterFadeOut();
void HandleEnterComplete();
void LeaveSequence(class UMenuSequence_TA* NextSequence);
void EnterSequence(class UMenuSequence_TA* PrevSequence);
};

// Class TAGame.MenuSequence_PremiumGarageReveal_TA
// 0x000C (0x00B4 - 0x00C0)
class UMenuSequence_PremiumGarageReveal_TA : public UMenuSequence_PremiumGarage_TA
{
public:
struct FName FinishCameraState; // 0x00B8 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.MenuSequence_PremiumGarageReveal_TA");
}

return uClassPointer;
};

void FinishReveal();
void LeaveSequence(class UMenuSequence_TA* NextSequence);
void EnterSequence(class UMenuSequence_TA* PrevSequence);
};

// Class TAGame.MenuSequence_PremiumGarageSnap_TA
// 0x0004 (0x00B4 - 0x00B8)
```

```

class UMenuSequence_PremiumGarageSnap_TA : public UMenuSequence_PremiumGarage_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuSequence_PremiumGarageSnap_TA");
}

return uClassPointer;
};

void LeaveSequence(class UMenuSequence_TA* NextSequence);
void EnterSequence(class UMenuSequence_TA* PrevSequence);
};

// Class TAGame.MenuTreePlaylistQueue_TA
// 0x0058 (0x0060 - 0x00B8)
class UMenuTreePlaylistQueue_TA : public UObject
{
public:
TArray<struct FActivePlaylistInfo> PlaylistData; // 0x0060 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FActivePlaylistQueueInfo> Queue; // 0x0070 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FActivePlaylistInfo> DebugPlaylistData; // 0x0080
(0x0010) [0x000100000400000] (CPF_NeedCtorLink)
class UIEpochNow* TimeProvider_Object; // 0x0090 (0x0008)
[0x0001000000000000]
class UIEpochNow* TimeProvider_Interface; // 0x0098 (0x0008)
[0x0001000000000000]
struct FScriptDelegate _EventActivePlaylistUpdated__Delegate; // 0x00A0
(0x0018) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MenuTreePlaylistQueue_TA");
}

return uClassPointer;
};

void SetDebugPlaylistData(TArray<struct FActivePlaylistInfo> InDebugPlaylistData);
static int32_t ActivePlaylistTimeSort(struct FActivePlaylistInfo A, struct FActivePlaylistInfo B);

```

```

void SetNextEpochTimer();
void HandleNextQueue();
TArray<struct FActivePlaylistQueueInfo> BuildPostQueue(uint64_t NewEntryEndTime);
void InsertIntoQueue(int32_t InDataIndex);
uint64_t GetNextQueueTransitionTime();
struct FActivePlaylistInfo GetActivePlaylistData();
int32_t GetCurrentQueueIndex();
void SetPlaylistData(TArray<struct FActivePlaylistInfo>& InPlaylistData);
void EventActivePlaylistUpdated(struct FActivePlaylistInfo NewActivePlaylist);
};

// Class TAGame.PointInSpace_TA
// 0x0010 (0x0268 - 0x0278)
class APointInSpace_TA : public AActor
{
public:
class USpriteComponent*           Sprite;           // 0x0268 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class UArrowComponent*           Arrow;           // 0x0270 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PointInSpace_TA");
}

return uClassPointer;
};

};

// Class TAGame.MirrorAxis_TA
// 0x0000 (0x0278 - 0x0278)
class AMirrorAxis_TA : public APointInSpace_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MirrorAxis_TA");
}

return uClassPointer;
};

```

```

void eventPostBeginPlay();
};

// Class TAGame.MtxErrors_TA
// 0x0010 (0x0080 - 0x0090)
class UMtxErrors_TA : public UErrorList
{
public:
class UErrorType*           MicrotransactionFulfillmentRestrictedWallet; // 0x0080
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType*           MicrotransactionFulfillmentRestrictedDaily; // 0x0088
(0x0008) [0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MtxErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.MultiltemDropsSave_TA
// 0x0010 (0x00C8 - 0x00D8)
class UMultiltemDropsSave_TA : public USaveObject_TA
{
public:
TArray<class UltemDropGroup_TA*>      QueuedFTEDropGroups;          // 0x00C8
(0x0010) [0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MultiltemDropsSave_TA");
}

return uClassPointer;
};

};

// Class TAGame.MusicTracklistNull_TA
// 0x0000 (0x0060 - 0x0060)

```

```

class UMusicTracklistNull_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.MusicTracklistNull_TA");
}

return uClassPointer;
};

class UIMusicTracklist_TA* Rebuild(class UMusicTracklistFactory_TA* Builder);
int32_t Num();
void NextTrack();
};

// Class TAGame.Mutator_Attachment_TA
// 0x0014 (0x0064 - 0x0078)
class UMutator_Attachment_TA : public UMutator_TA
{
public:
class ASpecialAttachment_TA* AttachmentArchetype; // 0x0068
(0x0008) [0x0000000000000001] (CPF_Edit)
class ASpecialAttachment_TA* Team1AttachmentArchetype; // 0x0070
(0x0008) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Attachment_TA");
}

return uClassPointer;
};

void SetupAttachment(class ACar_TA* Car);
void MutateObject(class UObject* O);
};

// Class TAGame.SpecialAttachment_TA
// 0x0038 (0x07A8 - 0x07E0)
class ASpecialAttachment_TA : public ARBActor_TA
{
public:

```

```

struct FVector Offset; // 0x07A8 (0x000C)
[0x0000000000000001] (CPF_Edit)
class UStaticMeshComponent* StaticMesh; // 0x07B8 (0x0008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
float BallHitMultiplier; // 0x07C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarHitMultiplier; // 0x07C4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarHitVerticalMultiplier; // 0x07C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarHitTorque; // 0x07CC (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bUseCarsBump : 1; // 0x07D0 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bDemolishOnHit : 1; // 0x07D0 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bDemolishTeam : 1; // 0x07D0 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
class ACar_TA* AttachedCar; // 0x07D8 (0x0008)
[0x0000000000002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialAttachment_TA");
}

return uClassPointer;
};

bool IsValidBump(float Speed, class ACar_TA* Car, struct FVector HitLocation, struct FVector HitNormal);
void OnHitCar(class ACar_TA* Car, struct FVector HitLocation, struct FVector HitNormal);
void OnHitBall(class ABall_TA* Ball, struct FVector HitLocation, struct FVector HitNormal);
void eventOnRigidBodyCollision(struct FAccumulatedRigidBodyCollision Collision);
void HandleDemolish(class ACar_TA* InCar, struct FDemolishData Data);
void AttachToCar(class ACar_TA* InCar, uint8_t InSocket);
void eventDestroyed();
};

// Class TAGame.Mutator_Audio_TA
// 0x000C (0x0064 - 0x0070)
class UMutator_Audio_TA : public UMutator_TA
{
public:
struct FName AudioSoundMode; // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Audio_TA");
}

return uClassPointer;
};

void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_Ball_TA
// 0x0064 (0x0064 - 0x00C8)
class UMutator_Ball_TA : public UMutator_TA
{
public:
class AGameEvent_Soccar_TA*           SoccarGame;           // 0x0068
(0x0008) [0x0000000000000000]
float                                BallScale;            // 0x0070 (0x0004)
[0x0000000000000000]
float                                BallBounciness;       // 0x0074 (0x0004)
[0x0000000000000000]
float                                BallCarBounciness;    // 0x0078 (0x0004)
[0x0000000000000000]
int32_t                             TotalGameBalls;      // 0x007C (0x0004)
[0x0000000000000000]
class ABall_TA*                      BallArchetype;       // 0x0080 (0x0008)
[0x0000000000000000]
float                                BallGravityScale;     // 0x0088 (0x0004)
[0x0000000000000000]
float                                BallSloMoDuration;   // 0x008C (0x0004)
[0x0000000000000000]
float                                BallSloMoDilation;   // 0x0090 (0x0004)
[0x0000000000000000]
float                                BallSloMoCooldown;   // 0x0094 (0x0004)
[0x0000000000000000]
float                                BallSloMoRadius;     // 0x0098 (0x0004)
[0x0000000000000000]
float                                BallSloMoDiffSpeed;  // 0x009C (0x0004)
[0x0000000000000000]
float                                BallMaxLinearSpeedScale; // 0x00A0 (0x0004)
[0x0000000000000000]
float                                BallMaxAngularSpeed; // 0x00A4 (0x0004)
[0x0000000000000000]
float                                BallHitSpinScale;    // 0x00A8 (0x0004)
[0x0000000000000000]
struct FVector                         BallMagnusCoefficient; // 0x00AC (0x000C)
[0x0000000000000000]
class UPhysicalMaterial*               PhysMatOverride;     // 0x00B8 (0x0008)
[0x0000000000000000]
float                                BallLiftImpulse;     // 0x00C0 (0x0004)

```

```

[0x0000000000000000]
float BallLiftDelay; // 0x00C4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Ball_TA");
}

return uClassPointer;
};

class UBallSlomoComponent_TA* GetSlomoComponent(class ABall_TA* Ball);
void MutateObject(class UObject* O);
void ApplyBallLift();
void HandleGameStateChanged(class AGameEvent_TA*_);
void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_Booster_TA
// 0x0018 (0x0064 - 0x007C)
class UMutator_Booster_TA : public UMutator_TA
{
public:
unsigned long bNoBooster : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bUnlimitedBoost : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bBoostRechargeGroundOnly : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long bOverrideStartBoost : 1; // 0x0068 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
float BoostStrength; // 0x006C (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostRechargeRate; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostRechargeDelay; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
float BoostStartAmount; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Booster_TA");
}

```

```
}

return uClassPointer;
};

void MutateObject(class UObject* O);
};

// Class TAGame.Mutator_Car_TA
// 0x0010 (0x0064 - 0x0074)
class UMutator_Car_TA : public UMutator_TA
{
public:
float CarScale; // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)
float JumpScale; // 0x006C (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarMaxLinearSpeedScale; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Car_TA");
}

return uClassPointer;
};

void MutateObject(class UObject* O);
};

// Class TAGame.Mutator_Demolish_TA
// 0x0006 (0x0064 - 0x006A)
class UMutator_Demolish_TA : public UMutator_TA
{
public:
uint8_t DemolishTarget; // 0x0068 (0x0001)
[0x0000000000000001] (CPF_Edit)
uint8_t DemolishSpeed; // 0x0069 (0x0001)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Demolish_TA");
}
```

```

}

return uClassPointer;
};

void MutateObject(class UObject* O);
};

// Class TAGame.Mutator_GameEvent_TA
// 0x001C (0x0064 - 0x0080)
class UMutator_GameEvent_TA : public UMutator_TA
{
public:
class AStatFactory_TA*           StatFactoryArchetype;          // 0x0068 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AAICController_TA*         BotArchetype;                  // 0x0070 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UEndRoundComponent_TA*     EndRoundComponentArchetype;    // 0x0078 (0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_GameEvent_TA");
}

return uClassPointer;
};

void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_GameSpeed_TA
// 0x0008 (0x0064 - 0x006C)
class UMutator_GameSpeed_TA : public UMutator_TA
{
public:
float                           GameSpeed;                   // 0x0068 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_GameSpeed_TA");
}
}

```

```
return uClassPointer;
};

void MutateObject(class UObject* O);
void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_Gravity_TA
// 0x0008 (0x0064 - 0x006C)
class UMutator_Gravity_TA : public UMutator_TA
{
public:
float Gravity; // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Gravity_TA");
}

return uClassPointer;
};

void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_Handicap_TA
// 0x0008 (0x0064 - 0x006C)
class UMutator_Handicap_TA : public UMutator_TA
{
public:
int32_t AutoBalanceDifference; // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Handicap_TA");
}

return uClassPointer;
};

void Init(class AGameEvent_TA* GameEvent);
```

```

};

// Class TAGame.Mutator_Loadout_TA
// 0x0014 (0x0064 - 0x0078)
class UMutator_Loadout_TA : public UMutator_TA
{
public:
TArray<class UProduct_TA*> Products; // 0x0068 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Loadout_TA");
}

return uClassPointer;
};

void __Mutator_Loadout_TA__Init_0x2(class AGameEvent_TA* _, class APRI_TA* PRI);
void __Mutator_Loadout_TA__Init_0x1(class AGameEvent_TA* _, class APRI_TA* PRI);
void HandleLoadoutRequested(struct FLoadoutData& OutLoadout, struct
FLoadoutAttributesArray& OutAttributes);
void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_Match_TA
// 0x0034 (0x0064 - 0x0098)
class UMutator_Match_TA : public UMutator_TA
{
public:
int32_t MatchGames; // 0x0068 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t GameTime; // 0x006C (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t FinishTime; // 0x0070 (0x0004)
[0x0000000000000001] (CPF_Edit)
int32_t GameScore; // 0x0074 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UGameEvent_Soccar_SubRules_TA* SubRules; // 0x0078
(0x0008) [0x0000000000000000]
unsigned long bUnlimitedTime : 1; // 0x0080 (0x0004)
[0x0000000000000000] [0x00000001]
class UOvertimeSettings_TA* OvertimeComponent; // 0x0088
(0x0008) [0x0000000000000001] (CPF_Edit)
class UMaxTimeComponent_TA* MaxTimeComponent; // 0x0090
(0x0008) [0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.Mutator_Match_TA");  
}  
  
return uClassPointer;  
};  
  
void Init(class AGameEvent_TA* GameEvent);  
};  
  
// Class TAGame.OvertimeSettings_TA  
// 0x0008 (0x0060 - 0x0068)  
class UOvertimeSettings_TA : public UObject  
{  
public:  
    uint8_t TieBreaker; // 0x0060 (0x0001)  
    [0x0000000000000001] (CPF_Edit)  
    int32_t OvertimeGameTime; // 0x0064 (0x0004)  
    [0x0000000000000001] (CPF_Edit)  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.OvertimeSettings_TA");  
        }  
  
        return uClassPointer;  
    };  
};  
  
// Class TAGame.Mutator_PhysicsLogPlayback_TA  
// 0x0004 (0x0064 - 0x0068)  
class UMutator_PhysicsLogPlayback_TA : public UMutator_TA  
{  
public:  
  
public:  
    static UClass* StaticClass()  
    {  
        static UClass* uClassPointer = nullptr;  
  
        if (!uClassPointer)  
        {  
            uClassPointer = UObject::FindClass("Class TAGame.Mutator_PhysicsLogPlayback_TA");  
        }  
    }  
};
```

```

return uClassPointer;
};

void __Mutator_PhysicsLogPlayback_TA__Init_0x1(class UGoal_TA* G);
void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.Mutator_RespawnTime_TA
// 0x0010 (0x0064 - 0x0074)
class UMutator_RespawnTime_TA : public UMutator_TA
{
public:
int32_t             RespawnTime;           // 0x0068 (0x0004)
[0x0000000000000000]
unsigned long        bDisableGoalDelay : 1; // 0x006C (0x0004)
[0x0000000000000000] [0x00000001]
float               BallRespawnTime;       // 0x0070 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Mutator_RespawnTime_TA");
}

return uClassPointer;
};

static void Mutate(class UMutator_RespawnTime_TA* Mutator, class UPlayerSpawnFeature_TA*
Feature);
void Init(class AGameEvent_TA* GameEvent);
};

// Class TAGame.PlayerSpawnFeature_TA
// 0x0010 (0x0060 - 0x0070)
class UPlayerSpawnFeature_TA : public UObject
{
public:
int32_t             RespawnDelaySeconds; // 0x0060 (0x0004)
[0x0000000000000000]
class AGameEvent_TA* GameEvent;          // 0x0068 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.PlayerSpawnFeature_TA");
}

return uClassPointer;
};

};

// Class TAGame.NameplateComponentMatinee_TA
// 0x0000 (0x00E0 - 0x00E0)
class UNameplateComponentMatinee_TA : public UNameplateComponent_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.NameplateComponentMatinee_TA");
}

return uClassPointer;
};

class FString GetPlayerName();
};

// Class TAGame.PartyClubSync_TA
// 0x0028 (0x0060 - 0x0088)
class UPartyClubSync_TA : public UObject
{
public:
class UOnlineGameParty_X*           Party;          // 0x0060 (0x0008)
[0x0001800000000000]
class UOnlineClubProvider_X*        Clubs;         // 0x0068 (0x0008)
[0x0001800000000000]
class UPsyNet_X*                   PsyNet;        // 0x0070 (0x0008)
[0x0001800000000000]
TArray<struct FUniqueNetId>        PlayerIds;    // 0x0078 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyClubSync_TA");
}
}

```

```

return uClassPointer;
};

bool __PartyClubSync_TA__GetPartyMemberIDs_0x2(struct FUniqueNetId Id);
struct FUniqueNetId __PartyClubSync_TA__GetPartyMemberIDs_0x1(struct FPartyMember P);
void ClearNonPartyClubData();
void SyncPlayerClub(struct FUniqueNetId PlayerID);
void HandlePartyClubChanged(class UOnlineMessageComponent_X* M, class
UPartyMessage_ClubChanged_TA* Message);
void HandleClubChanged(class UOnlineClubManager_X* M, uint64_t ClubID);
void HandlePartyChanged(class UOnlineGameParty_X* P);
TArray<struct FUniqueNetId> GetPartyMemberIDs();
void HandlePsyNetLogout();
void HandlePsyNetLogin();
void eventConstruct();
};

// Class TAGame.OnlineGameJoinGame_TA
// 0x0008 (0x0450 - 0x0458)
class UOnlineGameJoinGame_TA : public UOnlineGameJoinGame_X
{
public:
class UProductAssetCache_TA* AssetCache; // 0x0450 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameJoinGame_TA");
}

return uClassPointer;
};

TArray<int32_t> GetLoadout(struct FUniqueNetId PlayerID);
class UAddReservationMessage_X* AddLoadoutsToMessage(class
UAddReservationMessage_X* Message);
void ClearAssetCache();
void HandlePreLoadMap(class FString MapName);
void OnInit();
};

// Class TAGame.PartyMessage_Loadout_TA
// 0x0008 (0x00A8 - 0x00B0)
class UPartyMessage_Loadout_TA : public UPartyMessage_X
{
public:
class UPartyMemberLoadout_TA* MemberLoadout; // 0x00A8
(0x0008) [0x0000000000000000]

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_Loadout_TA");
}

return uClassPointer;
};

class UPartyMessage_Loadout_TA* SetLoadout(class UPartyMemberLoadout_TA*& InMemberLoadout);
};

// Class TAGame.PartyMessage_ProfileChange_TA
// 0x0058 (0x00A8 - 0x0100)
class UPartyMessage_ProfileChange_TA : public UPartyMessage_X
{
public:
struct FUniqueNetId           PlayerID;                      // 0x00A8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData>   ProfileProducts;        // 0x00F0
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_ProfileChange_TA");
}

return uClassPointer;
};

class UPartyMessage_ProfileChange_TA* SetProfileProducts(TArray<struct FOnlineProductData>& InProfileProducts);
class UPartyMessage_ProfileChange_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.PartyMessage_Rejoined_TA
// 0x0000 (0x00A8 - 0x00A8)
class UPartyMessage_Rejoined_TA : public UPartyMessage_X
{
public:

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_Rejoined_TA");
}

return uClassPointer;
};

};

// Class TAGame.PartyMessage_Chat_TA
// 0x0010 (0x00A8 - 0x00B8)
class UPartyMessage_Chat_TA : public UPartyMessage_X
{
public:
class FString Text; // 0x00A8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_Chat_TA");
}

return uClassPointer;
};

class UPartyMessage_Chat_TA* SetSender(struct FUniqueNetId InSender);
class UPartyMessage_Chat_TA* SetText(class FString InText);
};

// Class TAGame.PartyMessage_XPLevelUpdate_TA
// 0x004C (0x00A8 - 0x00F4)
class UPartyMessage_XPLevelUpdate_TA : public UPartyMessage_X
{
public:
struct FUniqueNetId PlayerID; // 0x00A8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t XpLevel; // 0x00F0 (0x0004)
[0x0001004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_XPLevelUpdate_TA");
}

return uClassPointer;
};

class UPartyMessage_XPLevelUpdate_TA* SetXPLevel(int32_t InXPLevel);
class UPartyMessage_XPLevelUpdate_TA* SetPlayer(struct FUniqueNetId InPlayerId);
};

// Class TAGame.OnlineGamePlaylists_TA
// 0x0000 (0x0118 - 0x0118)
class UOnlineGamePlaylists_TA : public UOnlineGamePlaylists_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGamePlaylists_TA");
}

return uClassPointer;
};

};

// Class TAGame.OnlineGameRegions_TA
// 0x0008 (0x0138 - 0x0140)
class UOnlineGameRegions_TA : public UOnlineGameRegions_X
{
public:
class URegionPingCache_TA*           RegionPingCache;          // 0x0138
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameRegions_TA");
}

return uClassPointer;
};

void OnInit();

```

```

};

// Class TAGame.RegionPingCache_TA
// 0x0018 (0x0060 - 0x0078)
class URegionPingCache_TA : public UObject
{
public:
TArray<struct FCachedRegionPing>           CachedRegionPings;          // 0x0060
(0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
class UOnlineGameRegions_TA*                 OnlineGameRegionsRef;      // 0x0070
(0x0008) [0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RegionPingCache_TA");
}

return uClassPointer;
};

class FString GetSaveFileName();
void HandleRegionsPinged(class UOnlineGameRegions_X* RegionsObj);
void HandleImportFinished(class ULocalCache_X* Cache, class URegionPingCache_TA*
CacheObject, class UError* Error);
void Init(class UOnlineGameRegions_TA* OnlineGameRegions);
};

// Class TAGame.OnlineGameReservations_TA
// 0x0010 (0x01C8 - 0x01D8)
class UOnlineGameReservations_TA : public UOnlineGameReservations_X
{
public:
TArray<struct FReservationLoadout>       Loadouts;                  // 0x01C8 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameReservations_TA");
}

return uClassPointer;
};

void __OnlineGameReservations_TA__OnInit_0x1(class AGameEvent_TA* _);

```

```

bool __OnlineGameReservations_TA__AddLoadout_0x2(int32_t Id);
int32_t __OnlineGameReservations_TA__AddLoadout_0x1(int32_t Id);
class UServerToServerMessage_X* GetMatchInfoMessage();
bool ShouldCheckPlatformRestrictions(class UAddReservationMessagePublic_X* Message);
bool CanAcceptReservations(class UAddReservationMessagePublic_X* Message);
TArray<int32_t> GetPrecacheProductIDs();
class UReservationsReadyMessage_X* CreateReadyMessage();
void ClearReservations();
void RemoveReservationIndex(int32_t Index);
void AddLoadout(struct FUniqueNetId PlayerID, TArray<int32_t> ProductIDs);
TArray<struct FReservationData> AddPlayersFromReservationMessage(class
UAddReservationMessage_X* ReservationMessage, class UIReservationConnection_X*
Connection);
class UObject* CreateClientReservationMessage(struct FServerReservationData& Reservation,
struct FUniqueNetId& PlayerID);
void HandleGameEventStarted();
void OnInit();
};

// Class TAGame.TourMatchInfoMessage_TA
// 0x0120 (0x0070 - 0x0190)
class UTourMatchInfoMessage_TA : public UServerToServerMessage_X
{
public:
    struct FTourServerSettings           TourSettings;           // 0x0070 (0x00E8)
    [0x0001000000400000] (CPF_NeedCtorLink)
    int32_t                           Playlist;               // 0x0158 (0x0004)
    [0x0001000000000000]
    class FString                      ReservationID;        // 0x0160 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)
    class FString                      JoinName;              // 0x0170 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)
    class FString                      JoinPassword;         // 0x0180 (0x0010)
    [0x0001000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.TourMatchInfoMessage_TA");
        }

        return uClassPointer;
    };

    struct FPsyNetBeaconReservation GetReservation();
};

// Class TAGame.TourPartyLeaderLeave_TA
// 0x0010 (0x0060 - 0x0070)
class UTourPartyLeaderLeave_TA : public UObject

```

```

{
public:
class UOnlineGameTournaments_TA*           Tournaments;          // 0x0060
(0x0008) [0x001004000002000] (CPF_Transient)
class UOnlineGameParty_X*                  Party;             // 0x0068 (0x0008)
[0x001004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourPartyLeaderLeave_TA");
}

return uClassPointer;
};

void __TourPartyLeaderLeave_TA__HandleLeaveTournamentMessage_0x1(class UGFxModal_X*
Modal);
class UGFxShell_TA* GetGFxShell();
void HandleLeaveTournamentConfirm(class UGFxModal_X* Modal);
void HandleLeaveTournamentMessage(class UOnlineMessageComponent_X* Component, class
UPartyMessage_TourLeaveTournament_TA* Message);
void HandleUnsubscribe(class UTourSubscriptions_TA* Subscriptions, uint64_t TournamentID);
void Init(class UOnlineGameTournaments_TA* InTournaments, class UOnlineGameParty_X*
InParty);
};

// Class TAGame.PsyNetService_CreateTournament_X
// 0x00F0 (0x00B0 - 0x01A0)
class UPsyNetService_CreateTournament_X : public UPsyNetService_ReservationBase_X
{
public:
int32_t                           Playlist;           // 0x00B0 (0x0004)
[0x0010000000000000]
struct FTourServerSettings         Settings;          // 0x00B8 (0x00E8)
[0x00100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_CreateTournament_X");
}

return uClassPointer;
};

```

```

struct FPsyNetBeaconReservation GetReservation();
};

// Class TAGame.PsyNetService_JoinTournament_X
// 0x0000 (0x00B0 - 0x00B0)
class UPsyNetService_JoinTournament_X : public UPsyNetService_ReservationBase_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_JoinTournament_X");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_CreateExternalMatch_X
// 0x00FC (0x00B0 - 0x01AC)
class UPsyNetService_CreateExternalMatch_X : public UPsyNetService_ReservationBase_X
{
public:
int32_t Playlist; // 0x00B0 (0x0004)
[0x0001000000000000]
struct FTourServerSettings Settings; // 0x00B8 (0x00E8)
[0x0001000000400000] (CPF_NeedCtorLink)
struct FFaceltReservationMatchInfo MatchInfo; // 0x01A0 (0x000C)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_CreateExternalMatch_X");
}

return uClassPointer;
};

struct FPsyNetBeaconReservation GetReservation();
};

// Class TAGame.PsyNetService_JoinExternalMatch_X
// 0x0000 (0x00B0 - 0x00B0)

```

```
class UPsyNetService_JoinExternalMatch_X : public UPsyNetService_ReservationBase_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_JoinExternalMatch_X");

}

return uClassPointer;
};

};

// Class TAGame.TourServerInfo_Facelt_TA
// 0x000C (0x0078 - 0x0084)
class UTourServerInfo_Facelt_TA : public UTourServerInfo_TA
{
public:
struct FFaceltReservationMatchInfo ReservationMatchInfo; // 0x0078
(0x000C) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourServerInfo_Facelt_TA");
}

return uClassPointer;
};

class UObject* CreateClientReservationMessage(class UReservationBeacon_X* Beacon, struct
FServerReservationData& Reservation, struct FUniqueNetId& PlayerID, struct
FTourServerSettings& TourSettings);
bool IsTourSet(struct FTourServerSettings& TourSettings);
};

// Class TAGame.OnlineGameVersion_TA
// 0x0004 (0x00B4 - 0x00B8)
class UOnlineGameVersion_TA : public UOnlineGameVersion_X
{
public:

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlineGameVersion_TA");
}

return uClassPointer;
};

void __OnlineGameVersion_TA__OnInit_0x1(class UOnlineGameTourMatchmaking_TA*
TourMatch);
void __OnlineGameVersion_TA__OnInit_0x2(class UOnlineGameTourMatchmaking_TA* _, class
UError* Error);
void OnInit();
};

// Class TAGame.OnlinePlayerAuthentication_TA
// 0x0008 (0x0208 - 0x0210)
class UOnlinePlayerAuthentication_TA : public UOnlinePlayerAuthentication_X
{
public:
class UFreeToPlayConfig_TA*           FreeToPlayConfig;           // 0x0208
(0x0008) [0x0000800000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.OnlinePlayerAuthentication_TA");
}

return uClassPointer;
};

void OnLoginSuccessRPC(class URPC_LoginAuthPlayer_X* RPC);
class UError* GetAuthLoginError();
void HandleFreeToPlayRequirementChanged();
class FString GetBuildRegion();
void OnInit();
};

// Class TAGame.RPC_ProductsClearExpired_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_ProductsClearExpired_TA : public URPC_X
{
public:
TArray<struct FProductInstanceId>      InstanceIDs;           // 0x00E8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
};

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsClearExpired_TA");
}

return uClassPointer;
};

class URPC_ProductsClearExpired_TA* SetInstanceIDs(TArray<struct FProductInstanceId> InIDs);
};

// Class TAGame.RPC_ProductsPlayerGet_TA
// 0x0088 (0x00E8 - 0x0170)
class URPC_ProductsPlayerGet_TA : public URPC_X
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData> ProductData; // 0x0130
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FOnlineProductData> ExpiredProducts; // 0x0140
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UOnlineProduct_TA*> Products; // 0x0150 (0x0010)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UOnlineProduct_TA*> ExpiredOnlineProducts; // 0x0160
(0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsPlayerGet_TA");
}

return uClassPointer;
};

void eventOnSuccess();
class URPC_ProductsPlayerGet_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_XPRequest_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_XPRequest_TA : public URPC_X
{

```

```
public:
struct FXPInfo           XPIInfoResponse;          // 0x00E8 (0x0020)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FUniqueNetId       PlayerID;                // 0x0108 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_XPRequest_TA");
}

return uClassPointer;
};

class URPC_XPRequest_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.PartyMessage_ClubChanged_TA
// 0x0008 (0x00A8 - 0x00B0)
class UPartyMessage_ClubChanged_TA : public UPartyMessage_X
{
public:
uint64_t                 ClubID;                  // 0x00A8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_ClubChanged_TA");
}

return uClassPointer;
};

class UPartyMessage_ClubChanged_TA* SetClubID(uint64_t InClubID);
};

// Class TAGame.PartyMessage_TourCheckInError_TA
// 0x0010 (0x00A8 - 0x00B8)
class UPartyMessage_TourCheckInError_TA : public UPartyMessage_X
{
public:
uint64_t                 TourID;                  // 0x00A8 (0x0008)
[0x0001000000000000]
struct FName              ErrorTypeName;           // 0x00B0 (0x0008)
```

```
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_TourCheckInError_TA");
}

return uClassPointer;
};

class UPartyMessage_TourCheckInError_TA* SetError(class UError* Error);
class UPartyMessage_TourCheckInError_TA* SetTourID(uint64_t InTourID);
};

// Class TAGame.PartyMessage_TourLeaveTournament_TA
// 0x0008 (0x00A8 - 0x00B0)
class UPartyMessage_TourLeaveTournament_TA : public UPartyMessage_X
{
public:
uint64_t           TourID;           // 0x00A8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_TourLeaveTournament_TA");
}

return uClassPointer;
};

class UPartyMessage_TourLeaveTournament_TA* SetTourID(uint64_t InTourID);
};

// Class TAGame.PartyMessage_TourPreCheckIn_TA
// 0x0008 (0x00A8 - 0x00B0)
class UPartyMessage_TourPreCheckIn_TA : public UPartyMessage_X
{
public:
class UTourSettings_TA*      Settings;      // 0x00A8 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_TourPreCheckIn_TA");
}

return uClassPointer;
};

class UPartyMessage_TourPreCheckIn_TA* SetTourSettings(class UTourSettings_TA*
InSettings);
};

// Class TAGame.PartyMessage_TourPreCheckInError_TA
// 0x0010 (0x00A8 - 0x00B8)
class UPartyMessage_TourPreCheckInError_TA : public UPartyMessage_X
{
public:
uint64_t TourID; // 0x00A8 (0x0008)
[0x0001000000000000]
struct FName ErrorTypeName; // 0x00B0 (0x0008)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PartyMessage_TourPreCheckInError_TA");
}

return uClassPointer;
};

class UPartyMessage_TourPreCheckInError_TA* SetError(class UErrorType* ErrorType);
class UPartyMessage_TourPreCheckInError_TA* SetTourID(uint64_t InTourID);
};

// Class TAGame.PlayerVanitySave_TA
// 0x0048 (0x00C8 - 0x0110)
class UPlayerVanitySave_TA : public USaveObject_TA
{
public:
struct FProductInstanceID InstanceID; // 0x00C8 (0x0010)
[0x0000000000000000]
int32_t ProductID; // 0x00D8 (0x0004)
[0x0000000000000000]
int32_t SelectedColorValue; // 0x00DC (0x0004)
[0x0008000000000000]
struct FScriptDelegate __EventVanityChanged__Delegate; // 0x00E0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```
struct FScriptDelegate           __SelectedColorValue__ChangeNotify;      // 0x00F8
(0x0018) [0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerVanitySave_TA");
}

return uClassPointer;
};

void __SelectedColorValue__ChangeNotifyFunc();
TArray<class UProductAttribute_TA*> GetAttributes();
void SetSelectedColorValue(int32_t InColor);
void SetProductInstance(struct FProductInstanceID InInstanceld, int32_t InProductID);
void GetDuplicateReplacementProduct(class USaveData_TA*& SaveData);
void Validate(class UProfile_TA* ForProfile);
void EventVanityChanged(struct FProductInstanceID InInstanceld);
class UProductSlot_TA* GetPlayerSlot();
int32_t GetDefaultProductID();
};

// Class TAGame.ProductAttribute_UserColorable_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_UserColorable_TA : public UProductAttribute_TA
{
public:
struct FLinearColor           DefaultColor;                      // 0x0080 (0x0010)
[0x0000000000000003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_UserColorable_TA");
}

return uClassPointer;
};

};

// Class TAGame.PlayerAvatarBorderSave_TA
// 0x0000 (0x0110 - 0x0110)
class UPlayerAvatarBorderSave_TA : public UPlayerVanitySave_TA
{
```

```
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerAvatarBorderSave_TA");
}

return uClassPointer;
};

class UProductSlot_TA* GetPlayerSlot();
int32_t GetDefaultProductID();
};

// Class TAGame.PlayerBannerSave_TA
// 0x0000 (0x0110 - 0x0110)
class UPlayerBannerSave_TA : public UPlayerVanitySave_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerBannerSave_TA");
}

return uClassPointer;
};

class UProductSlot_TA* GetPlayerSlot();
int32_t GetDefaultProductID();roductAttribute_UserColorable_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_UserColorable_TA : public UProductAttribute_TA
{
public:
struct FLinearColor DefaultColor; // 0x0080 (0x0010)
[0x0000000000000003] (CPF_Edit | CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
```

```

uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_UserColorable_TA");
}

return uClassPointer;
};

};

// Class TAGame.PlayerBindingUtils_TA
// 0x0000 (0x0060 - 0x0060)
class UPlayerBindingUtils_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerBindingUtils_TA");
}

return uClassPointer;
};

static void SetBindingsToPresetBindings(TArray<struct FPlayerBinding>& PlayerBindings,
TArray<struct FPlayerBinding>& PresetBindings, TArray<struct FPlayerBinding>&
DefaultBindings);
static void MergeLegacyBindings(TArray<struct FPlayerBinding>& PlayerBindings, TArray<struct
FPlayerBinding>& LegacyBindings, TArray<struct FPlayerBinding>& DefaultBindings);
static bool BindingsIdentical(TArray<struct FPlayerBinding>& Left, TArray<struct
FPlayerBinding>& Right);
static void CheckForNewBindings(TArray<struct FPlayerBinding> DefaultBindings, TArray<struct
FPlayerBinding>& Bindings);
static TArray<struct FPlayerBinding> MergeUserBindings(TArray<struct FPlayerBinding>&
DefaultBindings, TArray<struct FPlayerBinding>& UserBindings);
static void RemoveDefaultBindings(TArray<struct FPlayerBinding>& DefaultBindings,
TArray<struct FPlayerBinding>& Bindings);
static void RemoveDuplicateBindingIndexes(TArray<int32_t>& Indexes, TArray<struct
FPlayerBinding>& Bindings);
static TArray<int32_t> FindDuplicateBindingIndexes(TArray<struct FPlayerBinding>& Bindings);
static void SetDefaultRemappable(struct FName Action, TArray<struct FPlayerBinding>&
DefaultBindings, TArray<struct FPlayerBinding>& Bindings);
static void SetDefaultPressType(struct FName Action, TArray<struct FPlayerBinding>&
DefaultBindings, TArray<struct FPlayerBinding>& Bindings);
static void RemoveBinding(struct FName Action, TArray<struct FPlayerBinding>& Bindings);
static void ResetBinding(struct FName Action, TArray<struct FPlayerBinding>& DefaultBindings,
TArray<struct FPlayerBinding>& Bindings);
};

// Class TAGame.SpectatorVolume_TA
// 0x0004 (0x02A4 - 0x02A8)

```

```

class ASpectatorVolume_TA : public AVolume
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpectatorVolume_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_VoiceChatMatchToken_TA
// 0x0078 (0x0090 - 0x0108)
class UPsyNetService_VoiceChatMatchToken_TA : public UPsyNetClientService_X
{
public:
struct FUniqueNetId           PlayerID;                      // 0x0090 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 RoomId;                       // 0x00D8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 BaseUrl;                      // 0x00E8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString                 Token;                        // 0x00F8 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_VoiceChatMatchToken_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UProductMetrics_TA : public UMetricsGroup_X
{
public:

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductMetrics_TA");
}

return uClassPointer;
};

bool __ProductMetrics_TA__GetLoadoutMetricsData_0x2(struct FProductMetricsData P);
struct FOnlineProductAttribute __ProductMetrics_TA__GetLoadoutMetricsData_0x1(class UProductAttribute_TA* Attribute);
TArray<struct FProductMetricsData> GetLoadoutMetricsData(class ULoadout_TA* Loadout);
struct FPaintMetricsData GetPaintMetricsData(struct FLoadoutTeamPaint Paint);
void PlayWithLoadout(struct FUniqueNetId PlayerID, unsigned long bPrimaryPlayer, int32_t Team,
TArray<struct FProductMetricsData> Products, struct FPaintMetricsData Paint);
void RecordLoadout(class ULocalPlayer* Player, int32_t Team, class ULoadout_TA* Loadout);
void RecordLoadoutSet(class ULocalPlayer* Player, class UProfile_TA* Profile);
};

// Class TAGame.PlayerReportComponent_TA
// 0x0023 (0x009D - 0x00C0)
class UPlayerReportComponent_TA : public UActorComponent
{
public:
TArray<struct FUniqueNetId> ReportablePlayers; // 0x00A0 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FReportedPlayerInfo> ReportedPlayers; // 0x00B0
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerReportComponent_TA");
}

return uClassPointer;
};

void HandleDestroyed(class AGameEvent_TA* InGameEvent);
void AddReportedPlayer(struct FReportedPlayerInfo& ReportedPlayer);
void HandlePlayerAdded(class AGameEvent_TA* InGameEvent, class APRI_TA* PRI);
void eventConstruct();
};

// Class TAGame.PlayerInput_Game_TA

```

```

// 0x0000 (0x04C8 - 0x04C8)
class UPlayerInput_Game_TA : public UPlayerInput_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_Game_TA");
}

return uClassPointer;
};

void OnActiveBindingsChanged();
};

// Class TAGame.PlayerController_KnockOut_TA
// 0x0000 (0x0D008 - 0x0D008)
class APlayerController_KnockOut_TA : public APlayerController_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerController_KnockOut_TA");
}

return uClassPointer;
};

};

// Class TAGame.PlayerInputSequence_TA
// 0x0048 (0x00D0 - 0x0118)
class UPlayerInputSequence_TA : public UInteraction
{
public:

TArray<struct FName> Sequence; // 0x00D0 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) ConsoleCommand; // 0x00E0 (0x0010)
class FString ConsoleCommand;
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink) Timeout; // 0x00F0 (0x0004)
float Timeout;
[0x0000000000000001] (CPF_Edit) bOrderMatters : 1; // 0x00F4 (0x0004)
unsigned long bOrderMatters : 1;

```

```

[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bCheat : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bRemoveOnComplete : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
unsigned long bRemoveAfterSplashScreen : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000008] (CPF_Edit)
unsigned long bCaptureInput : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000010] (CPF_Edit)
unsigned long bConfirm : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000020] (CPF_Edit)
unsigned long bPrimaryPlayerOnly : 1; // 0x00F4 (0x0004)
[0x0000000000000001] [0x00000040] (CPF_Edit)
class FString Message; // 0x00F8 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<struct FName> CurrentSequence; // 0x0108 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerInputSequence_TA");
}

return uClassPointer;
};

void OnCommandConfirmed(class UGFxModal_X* Modal);
void RemoveSequence();
void OnSequenceCompleted();
void ResetSequence();
bool HandleFinalInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float AmountDepressed, unsigned long bGamepad);
bool HandleInputKey(int32_t ControllerId, struct FName Key, uint8_t EventType, float AmountDepressed, unsigned long bGamepad);
void SwapIndexes(struct FName A, struct FName B);
void Initialize();
bool ShouldGetInstance(class APlayerController* ForPC);
};

// Class TAGame.PlayerInput_Menu_TA
// 0x0010 (0x04C8 - 0x04D8)
class UPlayerInput_Menu_TA : public UPlayerInput_TA
{
public:
TArray<struct FName> AllowedActions; // 0x04C8 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()

```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_Menu_TA");  
}  
  
return uClassPointer;  
};  
  
void RemoveAllActionsBut(TArray<struct FName>& KeepActions, TArray<struct FPlayerBinding>& OutBindings);  
void OnActiveBindingsChanged();  
};  
  
// Class TAGame.PlayerInput_Edit_TA  
// 0x0000 (0x04C8 - 0x04C8)  
class UPlayerInput_Edit_TA : public UPlayerInput_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_Edit_TA");  
}  
  
return uClassPointer;  
};  
  
void OnActiveBindingsChanged();  
};  
  
// Class TAGame.PlayerInput_GameEditor_TA  
// 0x0000 (0x04C8 - 0x04C8)  
class UPlayerInput_GameEditor_TA : public UPlayerInput_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_GameEditor_TA");  
}
```

```
return uClassPointer;
};

void OnActiveBindingsChanged();
};

// Class TAGame.PlayerInput_Replay_TA
// 0x0000 (0x04C8 - 0x04C8)
class UPlayerInput_Replay_TA : public UPlayerInput_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_Replay_TA");
}

return uClassPointer;
};

void OnActiveBindingsChanged();
};

// Class TAGame.PlayerInput_Spectator_TA
// 0x0000 (0x04C8 - 0x04C8)
class UPlayerInput_Spectator_TA : public UPlayerInput_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerInput_Spectator_TA");
}

return uClassPointer;
};

void OnActiveBindingsChanged();
};

// Class TAGame.SeqEvent_InputSequence_TA
// 0x000C (0x017C - 0x0188)
class USeqEvent_InputSequence_TA : public USequenceEvent
{
```

```
public:
struct FName SequenceName; // 0x0180 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_InputSequence_TA");
}

return uClassPointer;
};

};

// Class TAGame.PlayerSpawnConfig_TA
// 0x0004 (0x0060 - 0x0064)
class UPlayerSpawnConfig_TA : public UObject
{
public:
int32_t RespawnDelaySeconds; // 0x0060 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PlayerSpawnConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.PlayerSpawnState_TA
// 0x0000 (0x0060 - 0x0060)
class UPlayerSpawnState_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
```

```

{
uClassPointer = UObject::FindClass("Class TAGame.PlayerSpawnState_TA");
}

return uClassPointer;
};

};

// Class TAGame.ReplicatedRespawnTicket_TA
// 0x0020 (0x0268 - 0x0288)
class AReplicatedRespawnTicket_TA : public AOwnerReplicatedActor_ORS
{
public:
float SecondsRemaining; // 0x0268 (0x0004)
[0x0000000000000000] (CPF_Net)
int32_t DisplaySeconds; // 0x026C (0x0004)
[0x0000000000000000]
struct FScriptDelegate _EventDisplaySecondsUpdated__Delegate; // 0x0270
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ReplicatedRespawnTicket_TA");
}

return uClassPointer;
};

void EventDisplaySecondsUpdated(int32_t NewDisplaySeconds);
};

// Class TAGame.SeqEvent_PreviewPostMatchCelebration_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_PreviewPostMatchCelebration_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.SeqEvent_PreviewPostMatchCelebration_TA");
}
}

```

```
return uClassPointer;
};

};

// Class TAGame.SeqEvent_MTXGarageReveal_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_MTXGarageReveal_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_MTXGarageReveal_TA");
}

return uClassPointer;
};

void SetRevealRarity(uint8_t InRarity);
};

// Class TAGame.StayAsPartyVoteYes_TA
// 0x0008 (0x0268 - 0x0270)
class AStayAsPartyVoteYes_TA : public AActor
{
public:
class AStayAsPartyVoter_TA* Voter; // 0x0268 (0x0008)
[0x0000000100002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StayAsPartyVoteYes_TA");
}

return uClassPointer;
};

void eventDestroyed();
void eventOnOwnerChanged();
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.LoadoutValidation_TA
```

```
// 0x0018 (0x0060 - 0x0078)
class ULoadoutValidation_TA : public UObject
{
public:
struct FScriptDelegate           __OnLoadoutCorrected__Delegate;      // 0x0060
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.LoadoutValidation_TA");
}

return uClassPointer;
};

static bool __LoadoutValidation_TA__RemoveIrrelevantAttributes_0x1(class
UProductAttribute_TA* A);
static void SetLoadoutSlotToDefaultID(int32_t TeamIndex, class UProductSlot_TA* Slot, struct
FLoadoutData& TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
static bool ValidateLoadoutTeamPaints(struct FUniqueNetId InPlayerId, struct FLoadoutData&
TeamLoadouts);
static bool ValidateLoadoutSlots(struct FUniqueNetId InPlayerId, struct FLoadoutData&
TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
static void ValidateLoadoutDLC(struct FUniqueNetId InPlayerId, struct FScriptDelegate
OnLoadoutModified, struct FScriptDelegate OnTeamPaintModified, struct FLoadoutData&
TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
static bool ShouldValidateOnlineProducts(struct FUniqueNetId InPlayerId);
static void CorrectDLCOwnership(class APRI_TA* InPRI, struct FScriptDelegate
OnLoadoutModified, struct FScriptDelegate OnTeamPaintModified);
static void RemoveCertifiedProductStat(struct FProductInstanceID InstanceID, TArray<struct
FOnlineProductStat>& OutOnlineProductStats);
static void RemoveIrrelevantAttributes(struct FLoadoutData& Loadout, struct
FLoadoutAttributesArray& OutLoadoutAttributes);
static bool ValidateOnlineProductWithLoadoutProduct(class UOnlineProduct_TA*
LoadoutOnlineProduct, int32_t LoadoutProductID, struct FUniqueNetId InPlayerId);
static bool ValidateTitleIDAttribute(class UProductAttribute_TitleID_TA* Current, class
UProductAttribute_TitleID_TA* Actual);
static bool ValidatePaintedAttribute(class UProductAttribute_Painted_TA* Current, class
UProductAttribute_Painted_TA* Actual);
static bool ValidateCertifiedAttribute(struct FOnlineProductStat ProductStat, class
UProductAttribute_Certified_TA* Actual);
static bool CorrectNonClientAuthoritativeAttributes(class UProductSlot_TA* Slot, struct
FLoadoutAttributesArray& OutAttributes, TArray<class UOnlineProduct_TA*>& PsyNetProducts);
static bool CorrectOnlineData(struct FUniqueNetId InPlayerId, class UProductsConfig_TA*
InProductsConfig, struct FLoadoutData& OutLoadout, struct FLoadoutAttributesArray&
OutAttributes, struct FClientLoadoutOnlineData& OutOnlineLoadout, TArray<struct
FOnlineProductStat>& OutOnlineProductStats, TArray<class UOnlineProduct_TA*>&
OnlineProducts);
static bool CorrectDuplicateAttributes(int32_t TeamIndex, class UProductSlot_TA* Slot, struct
```

```

FLoadoutAttributesArray& LoadoutAttributes);
static bool CorrectOfflineData(struct FUniqueNetId InPlayerId, struct FLoadoutData&
TeamLoadouts, struct FLoadoutAttributesArray& LoadoutAttributes);
void OnLoadoutCorrected();
};

// Class TAGame.RPC_ProductsLoadoutGet_TA
// 0x0088 (0x00E8 - 0x0170)
class URPC_ProductsLoadoutGet_TA : public URPC_X
{
public:
TArray<struct FInstanceToSlotRow> InstanceToSlotTable; // 0x00E8
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FUniqueNetId PlayerID; // 0x00F8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceId> Loadout; // 0x0140 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
TArray<struct FOnlineProductData> ProductData; // 0x0150
(0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<class UOnlineProduct_TA*> Products; // 0x0160 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsLoadoutGet_TA");
}

return uClassPointer;
};

void GetAllProductData(TArray<struct FOnlineProductData>& OutProductData);
void GetProducts(int32_t TeamIndex, TArray<class UOnlineProduct_TA*>& InLoadout);
void eventOnSuccess();
class URPC_ProductsLoadoutGet_TA* AddLoadout(int32_t TeamIndex, TArray<struct FProductInstanceId>& InLoadout);
class URPC_ProductsLoadoutGet_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.StayAsPartyLeader_TA
// 0x0048 (0x0060 - 0x00A8)
class UStayAsPartyLeader_TA : public UObject
{
public:
struct FUniqueNetId LeaderID; // 0x0060 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StayAsPartyLeader_TA");
}

return uClassPointer;
};

};

// Class TAGame.ServerInviteeConfirmedMergeEvent_TA
// 0x0000 (0x0060 - 0x0060)
class UServerInviteeConfirmedMergeEvent_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerInviteeConfirmedMergeEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.PRI_Breakout_TA
// 0x0000 (0x0BD8 - 0x0BD8)
class APRI_Breakout_TA : public APRI_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PRI_Breakout_TA");
}

return uClassPointer;
};

TArray<struct FScoreboardStat> GetScoreboardStats();
};

```

```
// Class TAGame.PrivacyPolicyConfig_TA
// 0x0018 (0x0078 - 0x0090)
class UPrivacyPolicyConfig_TA : public UOnlineConfig_X
{
public:
int32_t Version; // 0x0078 (0x0004)
[0x0000000000000000]
TArray<int32_t> IgnoredHashIds; // 0x0080 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PrivacyPolicyConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_AntennaPack_TA
// 0x0000 (0x00F8100 - 0x00F8100)
class UProductAsset_AntennaPack_TA : public UProductAsset_Pack_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_AntennaPack_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_CountryFlags_TA
// 0x0000 (0x00F8100 - 0x00F8100)
class UProductAsset_CountryFlags_TA : public UProductAsset_Pack_TA
{
public:

public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_CountryFlags_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_Currency_TA
// 0x0000 (0x00F8100 - 0x00F8100)
class UProductAsset_Currency_TA : public UProductAsset_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_Currency_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAsset_PlayerTitle_TA
// 0x0000 (0x00F8100 - 0x00F8100)
class UProductAsset_PlayerTitle_TA : public UProductAsset_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_PlayerTitle_TA");
}

return uClassPointer;
};
```

```
};

// Class TAGame.ProductAsset_UnderGlow_TA
// 0x0000 (0x01108 - 0x01108)
class UProductAsset_UnderGlow_TA : public UProductAsset_Attachment_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAsset_UnderGlow_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_BonusCrate_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_BonusCrate_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_BonusCrate_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductOverride_SkeletalMesh_TA
// 0x0008 (0x0060 - 0x0068)
class UProductOverride_SkeletalMesh_TA : public UProductOverride_TA
{
public:
class USkeletalMesh* Mesh; // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_SkeletalMesh_TA");
}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);

// Class TAGame.ProductOverride_StaticMesh_TA
// 0x0008 (0x0060 - 0x0068)
class UProductOverride_StaticMesh_TA : public UProductOverride_TA
{
public:
class UStaticMesh*           Mesh;           // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_StaticMesh_TA");
}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);

// Class TAGame.ProductOverride_Materials_TA
// 0x0010 (0x0060 - 0x0070)
class UProductOverride_Materials_TA : public UProductOverride_TA
{
public:
TArray<class UMaterialInterface*>      Materials;      // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_Materials_TA");
}
}

```

```

}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductAttribute_RocketPassPremium_TA
// 0x0000 (0x0080 - 0x0080)
class UProductAttribute_RocketPassPremium_TA : public UProductAttribute_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductAttribute_RocketPassPremium_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductAttribute_SwapMICParameterValues_TA
// 0x0010 (0x0080 - 0x0090)
class UProductAttribute_SwapMICParameterValues_TA : public UProductAttribute_TA
{
public:
TArray<struct FMICSwapParameters> Parameters; // 0x0080
(0x0010) [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductAttribute_SwapMICParameterValues_TA");
}

return uClassPointer;
};

void SwapParameterValue(class UMaterialInstanceConstant* MICToChange, struct FName
FirstParam, struct FName SecondParam, class UMeshComponent* ObjectToModify);
void ChangeValues(class UMeshComponent* ObjectToModify);

```

```
void ApplyToObject(class UProductAsset_TA* Asset, class UObject* Target);
};

// Class TAGame.ProductEquipProfileSlots_TA
// 0x0000 (0x0060 - 0x0060)
class UProductEquipProfileSlots_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductEquipProfileSlots_TA");
}

return uClassPointer;
};

};

// Class TAGame.ProductOverride_DisableSkelControl_TA
// 0x0008 (0x0060 - 0x0068)
class UProductOverride_DisableSkelControl_TA : public UProductOverride_TA
{
public:
struct FName           SkelControlToDisable;          // 0x0060 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_DisableSkelControl_TA");
}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductOverride_ParticleSystemComponent_TA
// 0x0008 (0x0060 - 0x0068)
class UProductOverride_ParticleSystemComponent_TA : public UProductOverride_TA
{
public:
class UParticleSystem*      ParticleSystem;          // 0x0060 (0x0008)
```

```
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_ParticleSystemComponent_TA");
}

return uClassPointer;
};

void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductOverride_StopAttachmentBehaviors_TA
// 0x0010 (0x0060 - 0x0070)
class UProductOverride_StopAttachmentBehaviors_TA : public UProductOverride_TA
{
public:
TArray<class UClass*> BehaviorsToZeroOut; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.ProductOverride_StopAttachmentBehaviors_TA");
}

return uClassPointer;
};

void __ProductOverride_StopAttachmentBehaviors_TA__ApplyToObject_0x1(class
UAttachmentBehavior_TA* A);
bool CanZeroOutCurrentAttachment(class UAttachmentBehavior_TA* AttachmentToCheck);
void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductOverride_WheelAttachment_TA
// 0x0010 (0x0060 - 0x0070)
class UProductOverride_WheelAttachment_TA : public UProductOverride_TA
{
public:
TArray<struct FWheelAttachment> AttachmentsToOverride; // 0x0060
(0x0010) [0x0000000000480001] (CPF_Edit | CPF_Component | CPF_NeedCtorLink)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_WheelAttachment_TA");
}

return uClassPointer;
};

class USkeletalMeshComponent* GetSkeletalMeshComponentOfWheel(class UObject* Target);
void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductOverride_WheelOverrides_TA
// 0x0020 (0x0060 - 0x0080)
class UProductOverride_WheelOverrides_TA : public UProductOverride_TA
{
public:
TArray<uint8_t> WheelsToChange; // 0x0060 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
TArray<class UProductOverride_TA*> OverridesToApply; // 0x0070
(0x0010) [0x0000000004400001] (CPF_Edit | CPF_NeedCtorLink | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductOverride_WheelOverrides_TA");
}

return uClassPointer;
};

void __ProductOverride_WheelOverrides_TA__ApplyOverridesToMeshComp_0x1(struct FAttachment X);
void ApplyOverridesToMeshComp(class UMeshComponent* MeshComp);
void ApplySideOverrides(class UMeshComponent* MeshComp, class UCarMeshComponent_TA* CarMeshComp);
void ApplyToObject(class UObject* Target);
};

// Class TAGame.ProductStat_SoccarEventBase_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_SoccarEventBase_TA : public UProductStat_TA
{
public:

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_SoccarEventBase_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
void HandleStatEvent(class APRI_TA* InPRI, class UStatEvent_TA* StatEvent, int32_t Count);
void ShutDown();
void OnInit();
};

// Class TAGame.ProductStat_AerialGoals_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_AerialGoals_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_AerialGoals_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_Assists_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Assists_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
```

```
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Assists_TA");  
}  
  
return uClassPointer;  
};  
  
void OnStatEvent(class UStatEvent_TA* StatEvent);  
};  
  
// Class TAGame.ProductStat_BackwardsGoals_TA  
// 0x0000 (0x00B0 - 0x00B0)  
class UProductStat_BackwardsGoals_TA : public UProductStat_SoccarEventBase_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_BackwardsGoals_TA");  
}  
  
return uClassPointer;  
};  
  
void OnStatEvent(class UStatEvent_TA* StatEvent);  
};  
  
// Class TAGame.ProductStat_Baskets_TA  
// 0x0000 (0x00B0 - 0x00B0)  
class UProductStat_Baskets_TA : public UProductStat_SoccarEventBase_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Baskets_TA");  
}  
  
return uClassPointer;  
};  
  
void OnStatEvent(class UStatEvent_TA* StatEvent);  
};
```

```

// Class TAGame.ProductStat_BicycleGoals_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_BicycleGoals_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_BicycleGoals_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_BoostTime_TA
// 0x0010 (0x00B0 - 0x00C0)
class UProductStat_BoostTime_TA : public UProductStat_TA
{
public:
class ACarComponent_Boost_TA*          BoostComponent;           // 0x00B0
(0x0008) [0x0000000000002000] (CPF_Transient)
float                  BoostStartTime;           // 0x00B8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                  LeftoverTime;           // 0x00BC (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_BoostTime_TA");
}

return uClassPointer;
};

static class FString GetDisplayValue(class APlayerControllerBase_TA* ForPC, int32_t StatValue);
void FinalizeTime();
void HandleActivationChanged(class ACarComponent_TA* InComponent);
void HandleBoostAdded(class ACarComponent_Boost_TA* InBoost);
void SetCar(class ACar_TA* InCar);
};

```

```
// Class TAGame.ProductStat_Centers_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Centers_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Centers_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_Clears_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Clears_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Clears_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_Demolishes_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Demolishes_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Demolishes_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_DistanceDriven_TA
// 0x000C (0x00B0 - 0x00BC)
class UProductStat_DistanceDriven_TA : public UProductStat_TA
{
public:
struct FVector OldLocation; // 0x00B0 (0x000C)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_DistanceDriven_TA");
}

return uClassPointer;
};

static FString GetDisplayValue(class APlayerControllerBase_TA* ForPC, int32_t StatValue);
void UpdateDistance();
void SetCar(class ACar_TA* InCar);
};

// Class TAGame.ProductStat_EpicSaves_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_EpicSaves_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_EpicSaves_TA");
}
}

```

```
return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_GamesPlayed_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_GamesPlayed_TA : public UProductStat_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_GamesPlayed_TA");
}

return uClassPointer;
};

void OnInit();
};

// Class TAGame.ProductStat_Goals_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Goals_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Goals_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_Juggles_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Juggles_TA : public UProductStat_SoccarEventBase_TA
{
```

```
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Juggles_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_LongGoals_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_LongGoals_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_LongGoals_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_MVPs_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_MVPs_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_MVPs_TA");
}
```

```
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_Saves_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Saves_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Saves_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_ShotsOnGoal_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_ShotsOnGoal_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_ShotsOnGoal_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_TimePlayed_TA
// 0x0000 (0x00B0 - 0x00B0)
```

```
class UProductStat_TimePlayed_TA : public UProductStat_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_TimePlayed_TA");
}

return uClassPointer;
};

static class FString GetDisplayValue(class APlayerControllerBase_TA* ForPC, int32_t StatValue);
void UpdateTime();
void SetCar(class ACar_TA* InCar);
};

// Class TAGame.ProductStat_TurtleGoals_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_TurtleGoals_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_TurtleGoals_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.ProductStat_Wins_TA
// 0x0000 (0x00B0 - 0x00B0)
class UProductStat_Wins_TA : public UProductStat_SoccarEventBase_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ProductStat_Wins_TA");
}

return uClassPointer;
};

void OnStatEvent(class UStatEvent_TA* StatEvent);
};

// Class TAGame.PsyNetService_CurrencyAdded_TA
// 0x0008 (0x0090 - 0x0098)
class UPsyNetService_CurrencyAdded_TA : public UPsyNetClientService_X
{
public:
int32_t CurrencyID; // 0x0090 (0x0004)
[0x0000000000000000] (CPF_Transient)
int32_t Amount; // 0x0094 (0x0004)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_CurrencyAdded_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_FaceltReservationFound_TA
// 0x0100 (0x0090 - 0x0190)
class UPsyNetService_FaceltReservationFound_TA : public UPsyNetClientService_X
{
public:
struct FCheckReservationResponse Reservation; // 0x0090
(0x0080) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
struct FFaceltMatchInfo MatchInfo; // 0x0110 (0x0080)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class
TAGame.PsyNetService_FaceltReservationFound_TA");
}

return uClassPointer;
};

struct FServerReservationData GetReservation();
};

// Class TAGame.PsyNetService_RewardDropReceived_TA
// 0x0070 (0x0090 - 0x0100)
class UPsyNetService_RewardDropReceived_TA : public UPsyNetClientService_X
{
public:
    struct FRocketPassInfo           RocketPassInfo;          // 0x0090 (0x000C)
    [0x0000000000000000]
    TArray<struct FOnlineProductData>   ProductData;          // 0x00A0
    (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FOnlineXPReward>      RewardDrops;         // 0x00B0
    (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FOnlineReward>        ChallengeRewards;    // 0x00C0
    (0x0010) [0x0000000000400000] (CPF_NeedCtorLink)
    TArray<struct FCurrency>           CurrencyDrops;       // 0x00D0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class FString                      Source;              // 0x00E0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)
    class FString                      MatchGuid;          // 0x00F0 (0x0010)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_RewardDropReceived_TA");
        }
    }

    return uClassPointer;
};

// Class TAGame.PsyNetService_TourConcluded_TA
// 0x0000 (0x0090 - 0x0090)
class UPsyNetService_TourConcluded_TA : public UPsyNetClientService_X
{
public:
    static UClass* StaticClass()
    {

```

```
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_TourConcluded_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_TourKicked_TA
// 0x0010 (0x0090 - 0x00A0)
class UPsyNetService_TourKicked_TA : public UPsyNetClientService_X
{
public:
uint64_t TournamentID; // 0x0090 (0x0008)
[0x0000000000000000]
class UErrorType* Reason; // 0x0098 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_TourKicked_TA");
}

return uClassPointer;
};

};

// Class TAGame.PsyNetService_TourMatchFound_TA
// 0x0030 (0x0090 - 0x00C0)
class UPsyNetService_TourMatchFound_TA : public UPsyNetClientService_X
{
public:
struct FTourMatch Match; // 0x0090 (0x0030)
[0x000100000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_TourMatchFound_TA");
}
}
```

```
return uClassPointer;
};

};

// Class TAGame.PsyNetService_TourReceivedBye_TA
// 0x0008 (0x0090 - 0x0098)
class UPsyNetService_TourReceivedBye_TA : public UPsyNetClientService_X
{
public:
    uint64_t TournamentID; // 0x0090 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.PsyNetService_TourReceivedBye_TA");
        }
    }

    return uClassPointer;
};

};

// Class TAGame.RemoteAvatarPermissions_TA
// 0x0000 (0x00F8 - 0x00F8)
class URemoteAvatarPermissions_TA : public URemoteAvatarPermissions_X
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RemoteAvatarPermissions_TA");
        }
    }

    return uClassPointer;
};

void OnAvatarPermissionSet(struct FUniqueNetId PlayerID, uint8_t PermissionStatus);
};

// Class TAGame.RequestPartyUpEvent_TA
// 0x0000 (0x0060 - 0x0060)
class URequestPartyUpEvent_TA : public UObject
```

```

{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RequestPartyUpEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.RewardDrop_XPMultiplier_TA
// 0x0004 (0x0068 - 0x006C)
class URewardDrop_XPMultiplier_TA : public URewardDrop_TA
{
public:
float Total; // 0x0068 (0x0004)
[0x000000040000000] (CPF_EditInlineNotify)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RewardDrop_XPMultiplier_TA");
}

return uClassPointer;
};

static class URewardDrop_XPMultiplier_TA* CreateInstance(struct FOnlineXPReward&
XPMultiplier);
};

// Class TAGame.RPC_RocketPassGetPlayerInfo_TA
// 0x0078 (0x00E8 - 0x0160)
class URPC_RocketPassGetPlayerInfo_TA : public URPC_X
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0001004000400000] (CPF_NeedCtorLink)
int32_t RocketPassID; // 0x0130 (0x0004)
[0x0001004000000000]
struct FRocketPassInfo RocketPassInfo; // 0x0134 (0x000C)
[0x0001000000000000]

```

```
struct FRocketPassStore           RocketPassStore;          // 0x0140 (0x0020)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RocketPassGetPlayerInfo_TA");
}

return uClassPointer;
};

class URPC_RocketPassGetPlayerInfo_TA* SetRocketPassID(int32_t InRocketPassID);
class URPC_RocketPassGetPlayerInfo_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_RocketPassGetRewardContent_TA
// 0x0030 (0x00E8 - 0x0118)
class URPC_RocketPassGetRewardContent_TA : public URPC_X
{
public:
int32_t                  RocketPassID;          // 0x00E8 (0x0004)
[0x0001004000000000]
int32_t                  TierCap;              // 0x00EC (0x0004)
[0x0001000000000000]
int32_t                  FreeMaxLevel;        // 0x00F0 (0x0004)
[0x0001000000000000]
int32_t                  PremiumMaxLevel;    // 0x00F4 (0x0004)
[0x0001000000000000]
TArray<struct FRocketPassRewardData>   FreeRewards;          // 0x00F8
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FRocketPassRewardData>   PremiumRewards;     // 0x0108
(0x0010) [0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RocketPassGetRewardContent_TA");
}

return uClassPointer;
};

class URPC_RocketPassGetRewardContent_TA* SetRocketPassID(int32_t InRocketPassID);
};
```

```

// Class TAGame.RPC_RocketPassGetPlayerPrestigeRewards_TA
// 0x0060 (0x00E8 - 0x0148)
class URPC_RocketPassGetPlayerPrestigeRewards_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    int32_t                      RocketPassID;       // 0x0130 (0x0004)
    [0x0001004000000000]
    TArray<struct FRocketPassRewardData>   PrestigeRewards;     // 0x0138
    (0x0010) [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class
TAGame.RPC_RocketPassGetPlayerPrestigeRewards_TA");
        }

        return uClassPointer;
    };

    class URPC_RocketPassGetPlayerPrestigeRewards_TA* SetRocketPassID(int32_t
InRocketPassID);
    class URPC_RocketPassGetPlayerPrestigeRewards_TA* SetPlayerID(struct FUniqueNetId
InPlayerId);
};

// Class TAGame.RPC_RocketPassPurchasePremium_TA
// 0x0088 (0x00E8 - 0x0170)
class URPC_RocketPassPurchasePremium_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    TArray<struct FProductInstanceId>   InstanceIDs;      // 0x0130 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)
    int32_t                      PurchasableID;       // 0x0140 (0x0004)
    [0x0001004000000000]
    unsigned long                 bOwnsPremium : 1;      // 0x0144 (0x0004)
    [0x0001000000002000] [0x00000001] (CPF_Transient)
    int32_t                      TierLevel;          // 0x0148 (0x0004)
    [0x0001000000002000] (CPF_Transient)
    TArray<struct FRocketPassUnlock>   Unlocks;          // 0x0150 (0x0010)
    [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FCurrency>         WalletCurrencies; // 0x0160 (0x0010)
    [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RocketPassPurchasePremium_TA");
}

return uClassPointer;
};

class URPC_RocketPassPurchasePremium_TA* SetPurchasable(int32_t InPurchasableID);
class URPC_RocketPassPurchasePremium_TA* SetKeys(TArray<struct FProductInstanceId>&
KeyInstanceIDs);
class URPC_RocketPassPurchasePremium_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_RocketPassPurchaseTiers_TA
// 0x0080 (0x00E8 - 0x0168)
class URPC_RocketPassPurchaseTiers_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0001004000400000] (CPF_NeedCtorLink)
TArray<struct FProductInstanceId> InstanceIDs;   // 0x0130 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
int32_t                      PurchasableID;     // 0x0140 (0x0004)
[0x0001004000000000]
int32_t                      TierLevel;         // 0x0144 (0x0004)
[0x0001000000002000] (CPF_Transient)
TArray<struct FRocketPassUnlock> Unlocks;        // 0x0148 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)
TArray<struct FCurrency>      WalletCurrencies; // 0x0158 (0x0010)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_RocketPassPurchaseTiers_TA");
}

return uClassPointer;
};

class URPC_RocketPassPurchaseTiers_TA* SetPurchasable(int32_t InPurchasableID);
class URPC_RocketPassPurchaseTiers_TA* SetKeys(TArray<struct FProductInstanceId>&
KeyInstanceIDs);
class URPC_RocketPassPurchaseTiers_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

```

```

// Class TAGame.RocketPassErrors_TA
// 0x0030 (0x0080 - 0x00B0)
class URocketPassErrors_TA : public UErrorList
{
public:
class UErrorType*           InvalidRocketPass;          // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           RocketPassNotActive;       // 0x0088 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           InvalidKey;                // 0x0090 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           PlayerAlreadyPremium;      // 0x0098 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           DoesNotOwnPremium;        // 0x00A0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           PurchasableNotFound;       // 0x00A8 (0x0008)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RocketPassErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_BanSelf_TA
// 0x0028 (0x00E8 - 0x0110)
class URPC_BanSelf_TA : public URPC_X
{
public:
int32_t                  DurationSeconds;           // 0x00E8 (0x0004)
[0x0000000000000000]
class FString              ReasonID;                 // 0x00F0 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString              BanType;                  // 0x0100 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_BanSelf_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.RPC_Challenge_ResetAllProgress_TA
// 0x0048 (0x00E8 - 0x0130)
class URPC_Challenge_ResetAllProgress_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_ResetAllProgress_TA");
}

return uClassPointer;
};

class URPC_Challenge_ResetAllProgress_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_Challenge_ResetChallengeProgress_TA
// 0x004C (0x00E8 - 0x0134)
class URPC_Challenge_ResetChallengeProgress_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t                         ChallengeID;     // 0x0130 (0x0004)
[0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_Challenge_ResetChallengeProgress_TA");
}

return uClassPointer;
};

```

```

class URPC_Challenge_ResetChallengeProgress_TA* SetChallengeID(int32_t InChallengeID);
class URPC_Challenge_ResetChallengeProgress_TA* SetPlayerID(struct FUniqueNetId
InPlayerId);
};

// Class TAGame.RPC_Challenge_SetProgress_TA
// 0x0050 (0x00E8 - 0x0138)
class URPC_Challenge_SetProgress_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001000000400000] (CPF_NeedCtorLink)
    int32_t                      ChallengeID;        // 0x0130 (0x0004)
    [0x0001000000000000]
    int32_t                      Count;              // 0x0134 (0x0004)
    [0x0001000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_Challenge_SetProgress_TA");
        }

        return uClassPointer;
    };

    class URPC_Challenge_SetProgress_TA* SetCount(int32_t InCount);
    class URPC_Challenge_SetProgress_TA* SetChallengeID(int32_t InChallengeID);
    class URPC_Challenge_SetProgress_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_DebugAddXP_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_DebugAddXP_TA : public URPC_X
{
public:
    int32_t                      XPToAdd;            // 0x00E8 (0x0004)
    [0x0000004000000000]
    struct FUniqueNetId           PlayerID;           // 0x00F0 (0x0048)
    [0x0000000000400000] (CPF_NeedCtorLink)
    struct FXPInfo                XPInfoResponse;     // 0x0138 (0x0020)
    [0x0000000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {

```

```

uClassPointer = UObject::FindClass("Class TAGame.RPC_DebugAddXP_TA");
}

return uClassPointer;
};

class URPC_DebugAddXP_TA* SetXPAmount(int32_t InXPToAdd);
class URPC_DebugAddXP_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_DebugSetXP_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_DebugSetXP_TA : public URPC_X
{
public:
int32_t XPToSet; // 0x00E8 (0x0004)
[0x0000004000000000]
struct FUniqueNetId PlayerID; // 0x00F0 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FXPInfo XPIInfoResponse; // 0x0138 (0x0020)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_DebugSetXP_TA");
}

return uClassPointer;
};

class URPC_DebugSetXP_TA* SetXPAmount(int32_t InXPToSet);
class URPC_DebugSetXP_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_DebugUnlinkPrimaryAccount_TA
// 0x0000 (0x00E8 - 0x00E8)
class URPC_DebugUnlinkPrimaryAccount_TA : public URPC_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_DebugUnlinkPrimaryAccount_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.RPC_DecaySkill_TA
// 0x0000 (0x00E8 - 0x00E8)
class URPC_DecaySkill_TA : public URPC_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_DecaySkill_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_FaceltMatchStarted_TA
// 0x0040 (0x00E8 - 0x0128)
class URPC_FaceltMatchStarted_TA : public URPC_X
{
public:
int32_t MatchID; // 0x00E8 (0x0004)
[0x0001000000000000]
class FString MatchGuid; // 0x00F0 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
class FString ServerId; // 0x0100 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t SeriesRoundNumber; // 0x0110 (0x0004)
[0x0001000000000000]
TArray<struct FUniqueNetId> Players; // 0x0118 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_FaceltMatchStarted_TA");
}

return uClassPointer;
}

```

```

};

class URPC_FaceltMatchStarted_TA* SetPlayerIDs(TArray<struct FUniqueNetId> Value);
class URPC_FaceltMatchStarted_TA* SetSeriesRoundNumber(int32_t Value);
class URPC_FaceltMatchStarted_TA* SetServerID(class FString Value);
class URPC_FaceltMatchStarted_TA* SetMatchGUID(class FString Value);
class URPC_FaceltMatchStarted_TA* SetMatchID(int32_t Value);
};

// Class TAGame.RPC_FaceltUpdateGame_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_FaceltUpdateGame_TA : public URPC_X
{
public:
int32_t MatchID; // 0x00E8 (0x0004)
[0x0001000000000000]
class FString ServerId; // 0x00F0 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t SeriesRoundNumber; // 0x0100 (0x0004)
[0x0001000000000000]
struct FTourMatchGame GameData; // 0x0108 (0x0018)
[0x0001000000000000]
uint64_t SeriesWinner; // 0x0120 (0x0008)
[0x0001000000000000]
TArray<struct FTourPlayerStats> PlayersStats; // 0x0128 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> PlayersNoShow; // 0x0138 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)
TArray<struct FUniqueNetId> PlayersLeftEarly; // 0x0148 (0x0010)
[0x0001000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_FaceltUpdateGame_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_GetSpecialEventCurrency_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_GetSpecialEventCurrency_TA : public URPC_SpecialEventBase_TA
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0001000000400000] (CPF_NeedCtorLink)
int32_t EventID; // 0x0130 (0x0004)

```

```
[0x0001000000000000]
struct FEventCurrencyData           EventCurrency;          // 0x0138 (0x0018)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetSpecialEventCurrency_TA");
}

return uClassPointer;
};

class URPC_GetSpecialEventCurrency_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
class URPC_GetSpecialEventCurrency_TA* SetEventID(int32_t InEventID);
};

// Class TAGame.RPC_GetSpecialEvents_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_GetSpecialEvents_TA : public URPC_SpecialEventBase_TA
{
public:
TArray<struct FSpecialEventData>      Events;          // 0x00E8 (0x0010)
[0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetSpecialEvents_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_GetWallet_TA
// 0x0058 (0x00E8 - 0x0140)
class URPC_GetWallet_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
[0x0000000000400000] (CPF_NeedCtorLink)
TArray<struct FCurrency>      Currencies;        // 0x0130 (0x0010)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_GetWallet_TA");
}

return uClassPointer;
};

class URPC_GetWallet_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_ProductsDebugAddProduct_TA
// 0x00B0 (0x00E8 - 0x0198)
class URPC_ProductsDebugAddProduct_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
int32_t                      ProductID;          // 0x0130 (0x0004)
[0x0000004000000000]
int32_t                      SeriesID;           // 0x0134 (0x0004)
[0x0000004000000000]
int32_t                      TradeHoldTimestamp; // 0x0138 (0x0004)
[0x0000004000000000]
TArray<struct FAddProductAttribute> Attributes; // 0x0140 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)
struct FOnlineProductData     Product;            // 0x0150 (0x0040)
[0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
class UOnlineProduct_TA*      OnlineProduct;       // 0x0190 (0x0008)
[0x0000004000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsDebugAddProduct_TA");
}

return uClassPointer;
};

void eventOnSuccess();
class URPC_ProductsDebugAddProduct_TA* SetSpecialEdition(int32_t SpecialEditionID);
class URPC_ProductsDebugAddProduct_TA* SetBlueprintCost(int32_t Cost);
class URPC_ProductsDebugAddProduct_TA* SetBlueprint(int32_t BlueprintID);
class URPC_ProductsDebugAddProduct_TA* SetPainted(int32_t PaintID);

```

```

class URPC_ProductsDebugAddProduct_TA* SetTeamEdition(int32_t TeamEditionId);
class URPC_ProductsDebugAddProduct_TA* SetCertified(int32_t CertifiedId);
class URPC_ProductsDebugAddProduct_TA* SetRentalLength(int32_t RentalLengthDays);
class URPC_ProductsDebugAddProduct_TA* SetExpirationTime(uint64_t ExpirationTime);
class URPC_ProductsDebugAddProduct_TA* SetTradeHoldTimeStamp(int32_t
InTradeHoldTimestamp);
class URPC_ProductsDebugAddProduct_TA* SetSeriesID(int32_t InSeriesID);
class URPC_ProductsDebugAddProduct_TA* SetProductID(int32_t InProductID);
class URPC_ProductsDebugAddProduct_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_ProductsDebugAddRandom_TA
// 0x0070 (0x00E8 - 0x0158)
class URPC_ProductsDebugAddRandom_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)
    int32_t                      Count;              // 0x0130 (0x0004)
    [0x0000004000000000]
    int32_t                      SeriesID;          // 0x0134 (0x0004)
    [0x0000004000000000]
    TArray<struct FOnlineProductData>   Drops;            // 0x0138 (0x0010)
    [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<class UOnlineProduct_TA*>     Products;          // 0x0148 (0x0010)
    [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsDebugAddRandom_TA");
        }

        return uClassPointer;
    };

    void eventOnSuccess();
    class URPC_ProductsDebugAddRandom_TA* SetSeriesID(int32_t InSeriesID);
    class URPC_ProductsDebugAddRandom_TA* SetCount(int32_t InCount);
    class URPC_ProductsDebugAddRandom_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_ProductsDebugSetCertified_TA
// 0x00C0 (0x00E8 - 0x01A8)
class URPC_ProductsDebugSetCertified_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)
    struct FProductInstanceId     InstanceID;         // 0x0130 (0x0010)

```

```

[0x0000004000000000] int32_t CertifiedStatID; // 0x0140 (0x0004)
[0x0000004000000000] int32_t CertifiedValue; // 0x0144 (0x0004)
[0x0000004000000000] struct FOnlineProductData Product; // 0x0148 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink) class UOnlineProduct_TA* OnlineProduct; // 0x0188 (0x0008)
[0x0000004000002000] (CPF_Transient) struct FScriptDelegate __EventCertifiedComplete__Delegate; // 0x0190
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsDebugSetCertified_TA");
}

return uClassPointer;
};

void eventOnComplete();
class URPC_ProductsDebugSetCertified_TA* NotifyCertifiedComplete(struct FScriptDelegate Callback);
class URPC_ProductsDebugSetCertified_TA* SetCertifiedValue(int32_t InCertifiedValue);
class URPC_ProductsDebugSetCertified_TA* SetCertifiedStatID(int32_t InCertifiedStatID);
class URPC_ProductsDebugSetCertified_TA* SetInstanceId(struct FProductInstanceId InInstanceld);
class URPC_ProductsDebugSetCertified_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
void EventCertifiedComplete(class UOnlineProduct_TA* NewProduct);
};

// Class TAGame.RPC_ProductsDebugSetEdition_TA
// 0x00C0 (0x00E8 - 0x01A8)
class URPC_ProductsDebugSetEdition_TA : public URPC_X
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink) struct FProductInstanceId InstanceID; // 0x0130 (0x0010)
[0x0000004000000000]
int32_t EditionID; // 0x0140 (0x0004)
[0x0000004000000000] struct FOnlineProductData Product; // 0x0148 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink) class UOnlineProduct_TA* OnlineProduct; // 0x0188 (0x0008)
[0x0000004000002000] (CPF_Transient) struct FScriptDelegate __EventEditionComplete__Delegate; // 0x0190
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsDebugSetEdition_TA");
}

return uClassPointer;
};

void eventOnComplete();
class URPC_ProductsDebugSetEdition_TA* NotifyEditionComplete(struct FScriptDelegate
Callback);
class URPC_ProductsDebugSetEdition_TA* SetEditionID(int32_t InEditionID);
class URPC_ProductsDebugSetEdition_TA* SetInstanceId(struct FProductInstanceId
InInstanceId);
class URPC_ProductsDebugSetEdition_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
void EventEditionComplete(class UOnlineProduct_TA* NewProduct);
};

// Class TAGame.RPC_ProductsDebugSetPainted_TA
// 0x00C0 (0x00E8 - 0x01A8)
class URPC_ProductsDebugSetPainted_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
struct FProductInstanceId     InstanceID;        // 0x0130 (0x0010)
[0x0000004000000000]
int32_t                      PaintID;          // 0x0140 (0x0004)
[0x0000004000000000]
struct FOnlineProductData    Product;          // 0x0148 (0x0040)
[0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
class UOnlineProduct_TA*      OnlineProduct;     // 0x0188 (0x0008)
[0x0000004000002000] (CPF_Transient)
struct FScriptDelegate        __EventPaintedComplete__Delegate; // 0x0190
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsDebugSetPainted_TA");
}

return uClassPointer;
};

```

```

void eventOnComplete();
class URPC_ProductsDebugSetPainted_TA* NotifyPaintedComplete(struct FScriptDelegate
Callback);
class URPC_ProductsDebugSetPainted_TA* SetPaintID(int32_t InPaintID);
class URPC_ProductsDebugSetPainted_TA* SetInstanceId(struct FProductInstanceID
InInstanceId);
class URPC_ProductsDebugSetPainted_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
void EventPaintedComplete(class UOnlineProduct_TA* NewProduct);
};

// Class TAGame.RPC_ProductsPlayerClear_TA
// 0x0048 (0x00E8 - 0x0130)
class URPC_ProductsPlayerClear_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_ProductsPlayerClear_TA");
        }

        return uClassPointer;
    }

    class URPC_ProductsPlayerClear_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_PsyNetGetVanities_TA
// 0x0040 (0x00E8 - 0x0128)
class URPC_PsyNetGetVanities_TA : public URPC_X
{
public:
    TArray<int32_t>           VanityTypes;           // 0x00E8 (0x0010)
    [0x0000004000400000] (CPF_NeedCtorLink)
    TArray<struct FUniqueNetId>       PlayerIds;       // 0x00F8 (0x0010)
    [0x0000004000400000] (CPF_NeedCtorLink)
    TArray<struct FVanityProductInfo>   VanityLoadouts;   // 0x0108
    (0x0010) [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FVanityProductData>   VanityProducts;   // 0x0118
    (0x0010) [0x0000004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
    }
}

```

```

{
uClassPointer = UObject::FindClass("Class TAGame.RPC_PsyNetGetVanities_TA");
}

return uClassPointer;
};

void eventOnSuccess();
class URPC_PsyNetGetVanities_TA* AddVanityType(int32_t InType);
class URPC_PsyNetGetVanities_TA* AddPlayerIDs(TArray<struct FUniqueNetId> InPlayerIDs);
class URPC_PsyNetGetVanities_TA* AddPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_PsyNetGetAvatars_TA
// 0x0000 (0x0128 - 0x0128)
class URPC_PsyNetGetAvatars_TA : public URPC_PsyNetGetVanities_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_PsyNetGetAvatars_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_PsyNetSetVanity_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_PsyNetSetVanity_TA : public URPC_X
{
public:
TArray<struct FVanityLoadout>           VanityLoadouts;          // 0x00E8 (0x0010)
[0x0000004000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_PsyNetSetVanity_TA");
}

return uClassPointer;
};

```

```

class URPC_PsyNetSetVanity_TA* AddVanityInfo(class UPlayerVanity_TA* Vanity, int32_t
InType);
};

// Class TAGame.RPC_RequestHonorDuel_TA
// 0x00B4 (0x00E8 - 0x019C)
class URPC_RequestHonorDuel_TA : public URPC_X
{
public:
    struct FUniqueNetId           Challenger;          // 0x00E8 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)
    struct FUniqueNetId           Defender;           // 0x0130 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)
    TArray<struct FUniqueNetId>   Spectators;        // 0x0178 (0x0010)
    [0x0000004000400000] (CPF_NeedCtorLink)
    class FString                 Region;             // 0x0188 (0x0010)
    [0x0000004000400000] (CPF_NeedCtorLink)
    int32_t                      BuildID;            // 0x0198 (0x0004)
    [0x0000004000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_RequestHonorDuel_TA");
        }

        return uClassPointer;
    }

    class URPC_RequestHonorDuel_TA* SetBuildID(int32_t InBuildID);
    class URPC_RequestHonorDuel_TA* SetRegion(class FString InRegion);
    class URPC_RequestHonorDuel_TA* SetSpectators(TArray<struct FUniqueNetId>& InSpectators);
    class URPC_RequestHonorDuel_TA* SetDefender(struct FUniqueNetId InDefender);
    class URPC_RequestHonorDuel_TA* SetChallenger(struct FUniqueNetId InChallenger);
};

// Class TAGame.RPC_SchematicResourcesGet_TA
// 0x004C (0x00E8 - 0x0134)
class URPC_SchematicResourcesGet_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0000004000400000] (CPF_NeedCtorLink)
    int32_t                      TotalSchematicResources; // 0x0130 (0x0004)
    [0x0000004000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_SchematicResourcesGet_TA");
}

return uClassPointer;
};

class URPC_SchematicResourcesGet_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_SubmitUserBugReport_TA
// 0x0080 (0x00E8 - 0x0168)
class URPC_SubmitUserBugReport_TA : public URPC_X
{
public:
struct FGuid [0x0001000000000000] AppSessionID; // 0x00E8 (0x0010)
struct FGuid [0x0001000000000000] LevelSessionID; // 0x00F8 (0x0010)
class FString [0x000100000400000] (CPF_NeedCtorLink) MatchGuid; // 0x0108 (0x0010)
int32_t [0x0001000000000000] MatchSeconds; // 0x0118 (0x0004)
class FString [0x000100000400000] (CPF_NeedCtorLink) BugDescription; // 0x0120 (0x0010)
TArray<struct FName> [0x000100000400000] (CPF_NeedCtorLink) BugCategories; // 0x0130 (0x0010)
unsigned long [0x0001000000000000] [0x00000001] bDescriptionFiltered : 1; // 0x0140 (0x0004)
class FString [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink) ConfirmationCode; // 0x0148 (0x0010)
class FString [0x000100000402000] (CPF_Transient | CPF_NeedCtorLink) UploadZipURL; // 0x0158 (0x0010)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_SubmitUserBugReport_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_TourGetMatchReservations_TA
// 0x00F8 (0x00E8 - 0x01E0)

```

```
class URPC_TourGetMatchReservations_TA : public URPC_X
{
public:
class FString ServerId; // 0x00E8 (0x0010)
[0x0001004000400000] (CPF_NeedCtorLink)
struct FTourServerSettings Settings; // 0x00F8 (0x00E8)
[0x0001000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetMatchReservations_TA");
}

return uClassPointer;
};

class URPC_TourGetMatchReservations_TA* SetServerID(class FString InServerID);
};

// Class TAGame.RPC_TourGetMatchReservations_Facelt_TA
// 0x0000 (0x01E0 - 0x01E0)
class URPC_TourGetMatchReservations_Facelt_TA : public
URPC_TourGetMatchReservations_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.RPC_TourGetMatchReservations_Facelt_TA");
}

return uClassPointer;
};

};

// Class TAGame.RPC_TourGetMatchServer_TA
// 0x0054 (0x00E8 - 0x013C)
class URPC_TourGetMatchServer_TA : public URPC_X
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
```

```

uint64_t TournamentID; // 0x0130 (0x0008)
[0x0000004000000000]
int32_t MatchID; // 0x0138 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetMatchServer_TA");
}

return uClassPointer;
};

class URPC_TourGetMatchServer_TA* SetMatchID(int32_t InMatchID);
class URPC_TourGetMatchServer_TA* SetTournamentID(uint64_t InTournamentID);
class URPC_TourGetMatchServer_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_TourGetRewards_TA
// 0x0060 (0x00E8 - 0x0148)
class URPC_TourGetRewards_TA : public URPC_X
{
public:
struct FUniqueNetId PlayerID; // 0x00E8 (0x0048)
[0x0001004000400000] (CPF_NeedCtorLink)
uint64_t TournamentID; // 0x0130 (0x0008)
[0x0001004000000000]
TArray<struct FTourReward> Rewards; // 0x0138 (0x0010)
[0x000100000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetRewards_TA");
}

return uClassPointer;
};

class URPC_TourGetRewards_TA* SetTournamentID(uint64_t InTournamentID);
class URPC_TourGetRewards_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_TourGetSubscriptions_TA
// 0x00E8 (0x00E8 - 0x01D0)

```

```

class URPC_TourGetSubscriptions_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    TArray<class UTourSettings_TA*>     Tournaments;      // 0x0130
    (0x0010) [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<uint64_t>             CreatorOf;        // 0x0140 (0x0010)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<uint64_t>             AdminOf;          // 0x0150 (0x0010)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<uint64_t>             Registered;       // 0x0160 (0x0010)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    uint64_t                     ActiveTournamentID; // 0x0170 (0x0008)
    [0x0001004000002000] (CPF_Transient)
    struct FTourTeam              ActiveTeam;        // 0x0178 (0x0048)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<struct FTourResult>    Results;          // 0x01C0 (0x0010)
    [0x0001004000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourGetSubscriptions_TA");
        }

        return uClassPointer;
    };

    class URPC_TourGetSubscriptions_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_TourStartMatchmaking_TA
// 0x0048 (0x00E8 - 0x0130)
class URPC_TourStartMatchmaking_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourStartMatchmaking_TA");
        }
    }
}

```

```

return uClassPointer;
};

class URPC_TourStartMatchmaking_TA* SetPlayerID(struct FUniqueNetId& InPlayerId);
};

// Class TAGame.RPC_TourUnsubscribe_TA
// 0x0068 (0x00E8 - 0x0150)
class URPC_TourUnsubscribe_TA : public URPC_X
{
public:
    struct FUniqueNetId           PlayerID;           // 0x00E8 (0x0048)
    [0x0001004000400000] (CPF_NeedCtorLink)
    uint64_t                      TournamentID;      // 0x0130 (0x0008)
    [0x0001004000000000]
    unsigned long                 bUnsubscribeAnyRegisteredTournament : 1; // 0x0138
    (0x0004) [0x0001004000000000] [0x00000001]
    TArray<struct FUniqueNetId>   TeamMembers;        // 0x0140 (0x0010)
    [0x0001004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.RPC_TourUnsubscribe_TA");
        }
    }

    return uClassPointer;
};

class URPC_TourUnsubscribe_TA* SetTeamMembers(TArray<struct FUniqueNetId>
InTeamMembers);
class URPC_TourUnsubscribe_TA* SetUnsubscribeAnyRegisteredTournament(unsigned long
bValue);
class URPC_TourUnsubscribe_TA* SetTournamentID(uint64_t InTournamentID);
class URPC_TourUnsubscribe_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.RPC_UpdateTwoFactorStatus_TA
// 0x0010 (0x00E8 - 0x00F8)
class URPC_UpdateTwoFactorStatus_TA : public URPC_X
{
public:
    class FString                EpicAuthTicket;     // 0x00E8 (0x0010)
    [0x0000004000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
    }
};

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_UpdateTwoFactorStatus_TA");
}

return uClassPointer;
};

class URPC_UpdateTwoFactorStatus_TA* SetEpicAuthTicket(class FString InEpicAuthTicket);
};

// Class TAGame.RPC_UploadXP_TA
// 0x0050 (0x00E8 - 0x0138)
class URPC_UploadXP_TA : public URPC_X
{
public:
struct FUniqueNetId           PlayerID;          // 0x00E8 (0x0048)
[0x0000004000400000] (CPF_NeedCtorLink)
float                          XPTotal;          // 0x0130 (0x0004)
[0x0000004000000000]
int32_t                        XpLevel;          // 0x0134 (0x0004)
[0x0000004000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.RPC_UploadXP_TA");
}

return uClassPointer;
};

class URPC_UploadXP_TA* SetXPLevel(int32_t InXPLevel);
class URPC_UploadXP_TA* SetXPTotal(float InXPTotal);
class URPC_UploadXP_TA* SetPlayerID(struct FUniqueNetId InPlayerId);
};

// Class TAGame.SaveDataMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class USaveDataMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.SaveDataMetrics_TA");
}

return uClassPointer;
};

void ReconcileXP(int32_t OnlineXP, int32_t LocalXP);
void RecordReconcileXP(int32_t OnlineXP, int32_t LocalXP);
};

// Class TAGame.Scout_TA
// 0x0000 (0x05B0 - 0x05B0)
class AScout_TA : public AScout
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Scout_TA");
}

return uClassPointer;
};

};

// Class TAGame.SketchedTextureComponent_TA
// 0x0034 (0x00A4 - 0x00D8)
class USketchedTextureComponent_TA : public UActorComponent_X
{
public:
struct FVector Resolution; // 0x00A8 (0x00C)
[0x0000000000000001] (CPF_Edit)
float Height; // 0x00B4 (0x004)
[0x0000000000000001] (CPF_Edit)
unsigned long bFaceCamera : 1; // 0x00B8 (0x004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class UStaticMeshComponent* Mesh; // 0x00C0 (0x008)
[0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
class USketchedTexture* TextureTarget; // 0x00C8 (0x008)
[0x0000004000002000] (CPF_Transient)
class APlayerController* Viewer; // 0x00D0 (0x008)
[0x0000000000000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SketchedTextureComponent_TA");
}

return uClassPointer;
};

void FaceViewer();
void eventTick(float DeltaTime);
void eventDetached();
void eventAttached();
};

// Class TAGame.SeqAct_AddGameBall_TA
// 0x0038 (0x0160 - 0x0198)
class USeqAct_AddGameBall_TA : public USequenceAction
{
public:
class ABall_TA*           SpawnedBall;          // 0x0160 (0x0008)
[0x0000000000000000]
class AActor*              Instigator;          // 0x0168 (0x0008)
[0x0000000000000000]
class AActor*              SpawnTransform;        // 0x0170 (0x0008)
[0x0000000000000000]
class AActor*              PlacedBall;          // 0x0178 (0x0008)
[0x0000000000000000]
unsigned long               bWake : 1;            // 0x0180 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class FString              BallArchetype;        // 0x0188 (0x0010)
[0x00008000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_AddGameBall_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_ApplyCarProducts_TA
// 0x0048 (0x0160 - 0x01A8)
class USeqAct_ApplyCarProducts_TA : public USequenceAction
{
public:
TArray<class UProductAsset_TA*>      Assets;          // 0x0160 (0x0010)

```

```

[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FLinearColor TeamColor; // 0x0170 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor AccentColor; // 0x0180 (0x0010)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_PaintFinish_TA* TeamFinish; // 0x0190
(0x0008) [0x0000000000000001] (CPF_Edit)
class UProductAsset_PaintFinish_TA* CustomFinish; // 0x0198
(0x0008) [0x0000000000000001] (CPF_Edit)
class ACarPreviewActor_TA* CarPreviewActor; // 0x01A0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ApplyCarProducts_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_AttachAssetToCar_TA
// 0x0020 (0x0160 - 0x0180)
class USeqAct_AttachAssetToCar_TA : public USequenceAction
{
public:
class UStaticMesh* StaticMeshToAttach; // 0x0160 (0x0008)
[0x0000000000000001] (CPF_Edit)
class USkeletalMesh* SkeletalMeshToAttach; // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UParticleSystem* ParticleSystemToAttach; // 0x0170
(0x0008) [0x0000000000000001] (CPF_Edit)
struct FName BoneToAttachTo; // 0x0178 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_AttachAssetToCar_TA");
}

return uClassPointer;
};

```

```

void eventActivated();
};

// Class TAGame.SeqAct_CheckRotationAngle_TA
// 0x0045 (0x0178 - 0x01BD)
class USeqAct_CheckRotationAngle_TA : public USeqAct_Latent
{
public:
class AActor* Actor; // 0x0178 (0x0008)
[0x0000000000000000]
class AActor* CheckActor; // 0x0180 (0x0008)
[0x0000000000000000]
float RotationAngle; // 0x0188 (0x0004)
[0x0000000000000000]
struct FRotator LastRotation; // 0x018C (0x000C)
[0x0000000000000000]
float LastDirection; // 0x0198 (0x0004)
[0x0000000000000000]
struct FVector CheckAxis; // 0x019C (0x000C)
[0x0000000000000000]
struct FVector TurnAxis; // 0x01A8 (0x000C)
[0x0000000000000000]
float MaxRotationAngle; // 0x01B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bResetOnDirectionChange : 1; // 0x01B8 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
uint8_t AxisDirection; // 0x01BC (0x0001)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_CheckRotationAngle_TA");
}

return uClassPointer;
};

struct FVector GetTurnAxis();
struct FVector GetCheckAxis();
bool eventUpdate(float DeltaTime);
void eventActivated();
};

// Class TAGame.SeqAct_CompleteFTECheckpoint_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_CompleteFTECheckpoint_TA : public USequenceAction
{
public:

```

```
class FString CheckpointName; // 0x0160 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_CompleteFTECheckpoint_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_ConcatenateStrings_TA
// 0x0038 (0x0160 - 0x0198)
class USeqAct_ConcatenateStrings_TA : public USequenceAction
{
public:
class FString ValueA; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString ValueB; // 0x0170 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long ConcatenateWithSpace : 1; // 0x0180 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class FString StringResult; // 0x0188 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ConcatenateStrings_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_ControlCarMovement_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_ControlCarMovement_TA : public USequenceAction
{
public:
```

```
class AActor*           Instigator;          // 0x0160 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ControlCarMovement_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_Counter_TA
// 0x0004 (0x0160 - 0x0164)
class USeqAct_Counter_TA : public USequenceAction
{
public:
int32_t           MaxCount;          // 0x0160 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_Counter_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_CreateAttachComponent_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_CreateAttachComponent_TA : public USequenceAction
{
public:
class AActor*           AttachTo;          // 0x0160 (0x0008)
[0x0000000000000000]
class UActorComponent_X*      AttachingComponent;    // 0x0168
(0x0008) [0x000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_CreateAttachComponent_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_DemoCar_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_DemoCar_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_DemoCar_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_DisableForceInFXActor_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_DisableForceInFXActor_TA : public USequenceAction
{
public:
class AFXActor_X*           InFXActor;          // 0x0160 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_DisableForceInFXActor_TA");
}
```

```

}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_DisplayMessage_TA
// 0x0050 (0x0160 - 0x01B0)
class USeqAct_DisplayMessage_TA : public USequenceAction
{
public:
    class AActor*           Instigator;          // 0x0160 (0x0008)
    [0x0000000000000000]
    class FString            Title;              // 0x0168 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString            Body;               // 0x0178 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    unsigned long             bShowButtonOne : 1; // 0x0188 (0x0004)
    [0x0000000000000001] [0x00000001] (CPF_Edit)
    unsigned long             bShowButtonTwo : 1; // 0x0188 (0x0004)
    [0x0000000000000001] [0x00000002] (CPF_Edit)
    class FString            ButtonOneTitle;      // 0x0190 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    class FString            ButtonTwoTitle;      // 0x01A0 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_DisplayMessage_TA");
        }
    }

    return uClassPointer;
};

void HandleButtonTwoClick(class UGFxModal_X* Modal);
void HandleButtonOneClick(class UGFxModal_X* Modal);
void DisplayMessage(class APlayerController_TA* PC);
void eventActivated();
};

// Class TAGame.SeqAct_DriveSpline_TA
// 0x0028 (0x0178 - 0x01A0)
class USeqAct_DriveSpline_TA : public USeqAct_Latent
{
public:
    float                  ForwardOffset;       // 0x0178 (0x0004)
    [0x0000000000000001] (CPF_Edit)
}

```

```
class ASplineActor* Spline; // 0x0180 (0x0008)
[0x0000000000000001] (CPF_Edit)
float Throttle; // 0x0188 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bDebug : 1; // 0x018C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
TArray<struct FCarSplineData> Cars; // 0x0190 (0x0010)
[0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_DriveSpline_TA");
}

return uClassPointer;
};

float GetSteer(class ACar_TA* Car, struct FVector Destination);
struct FCarSplineData DriveSpline(struct FCarSplineData CarData, float DeltaTime);
bool eventUpdate(float DeltaTime);
void eventActivated();
};

// Class TAGame.SeqAct_DynamicMapEvents_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_DynamicMapEvents_TA : public USequenceAction
{
public:
class FString EventID; // 0x0160 (0x0010)
[0x000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_DynamicMapEvents_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_EndTutorial_TA
// 0x0008 (0x0160 - 0x0168)
```

```
class USeqAct_EndTutorial_TA : public USequenceAction
{
public:
    class AActor* Instigator; // 0x0160 (0x0008)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_EndTutorial_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
}

// Class TAGame.SeqAct_ExitToMainMenu_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_ExitToMainMenu_TA : public USequenceAction
{
public:

    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ExitToMainMenu_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
}

// Class TAGame.SeqAct_ExplodeGameBall_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_ExplodeGameBall_TA : public USequenceAction
{
public:

    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;
```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ExplodeGameBall_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_GameURLHasOption_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_GameURLHasOption_TA : public USequenceAction
{
public:
class FString Option; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GameURLHasOption_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_GetDetailMode_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_GetDetailMode_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetDetailMode_TA");
}

return uClassPointer;
};
```

```
void eventActivated();
};

// Class TAGame.SeqAct_GetEdition_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_GetEdition_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetEdition_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_GetEffectIntensity_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_GetEffectIntensity_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetEffectIntensity_TA");
}

return uClassPointer;
};

static class APlayerControllerBase_TA* GetPrimaryPlayerController();
void eventActivated();
};

// Class TAGame.SeqAct_GetFTEState_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_GetFTEState_TA : public USequenceAction
{
public:
```

```
class FString CheckpointName; // 0x0160 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bIsActive : 1; // 0x0170 (0x0004)
[0x0000000000000000] [0x00000001]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetFTEState_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_GetGameBall_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_GetGameBall_TA : public USequenceAction
{
public:
class UObject* Ball; // 0x0160 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetGameBall_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_GetPlayerCar_TA
// 0x0018 (0x0160 - 0x0178)
class USeqAct_GetPlayerCar_TA : public USequenceAction
{
public:
class FString PlayerName; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class AActor* FoundActor; // 0x0170 (0x0008)
[0x0000000000002000] (CPF_Transient)
```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetPlayerCar_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_GetPlaylistType_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_GetPlaylistType_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetPlaylistType_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_GetRarityColor_TA
// 0x002C (0x0160 - 0x018C)
class USeqAct_GetRarityColor_TA : public USequenceAction
{
public:
TArray<struct FVector> RarityColorsPrime; // 0x0160 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
unsigned long bRandomize : 1; // 0x0170 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
uint8_t MinimumRandomQuality; // 0x0174 (0x0001)
[0x0000000000000001] (CPF_Edit)
int32_t ColorIdx; // 0x0178 (0x0004)
[0x0000000000000000]
struct FVector OutColor; // 0x017C (0x000C)
[0x0000000000002000] (CPF_Transient)

```

```
int32_t LastRandomIdx; // 0x0188 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetRarityColor_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_GetSaveData_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_GetSaveData_TA : public USequenceAction
{
public:
class USaveData_TA* SaveData; // 0x0160 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetSaveData_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_GetTeamScore_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_GetTeamScore_TA : public USequenceAction
{
public:
int32_t TeamIndex; // 0x0160 (0x0004)
[0x0000000000000000]
int32_t TeamScore; // 0x0164 (0x0004)
[0x0000000000000000]

public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetTeamScore_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_GetTutorialType_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_GetTutorialType_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetTutorialType_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_GetWeatherEnabled_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_GetWeatherEnabled_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GetWeatherEnabled_TA");
}

return uClassPointer;
};
```

```
};

void eventActivated();
};

// Class TAGame.SeqAct_GiveBoost_TA
// 0x000C (0x0160 - 0x016C)
class USeqAct_GiveBoost_TA : public USequenceAction
{
public:
    class AActor*           Instigator;          // 0x0160 (0x0008)
    [0x0000000000000000]
    float                   BoostAmount;         // 0x0168 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_GiveBoost_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
};

// Class TAGame.SeqAct_HighlightReelFinished_TA
// 0x0018 (0x0160 - 0x0178)
class USeqAct_HighlightReelFinished_TA : public USequenceAction
{
public:
    struct FScriptDelegate      _EventHighlightReelFinished__Delegate;    // 0x0160
    (0x0018) [0x00000000400000] (CPF_NeedCtorLink)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_HighlightReelFinished_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
    void EventHighlightReelFinished(class USeqAct_HighlightReelFinished_TA* Act);
```

```
};

// Class TAGame.SeqAct_InstanceMaterial_TA
// 0x0024 (0x0160 - 0x0184)
class USeqAct_InstanceMaterial_TA : public USequenceAction
{
public:
    class UMaterialInstanceConstant*          MaterialInstance;           // 0x0160
    (0x0008) [0x0000000000000001] (CPF_Edit)
    class ASkeletalMeshActor*                 SK;                      // 0x0168 (0x0008)
    [0x0000000000000000]
    class AStaticMeshActor*                  SM;                      // 0x0170 (0x0008)
    [0x0000000000000000]
    class UMaterialInstanceConstant*          InstancedMaterial;        // 0x0178
    (0x0008) [0x0000000000000000]
    int32_t                                MatIndex;                // 0x0180 (0x0004)
    [0x0000000000000000]

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_InstanceMaterial_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
};

// Class TAGame.SeqAct_IntroMenuCarsInPosition_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_IntroMenuCarsInPosition_TA : public USequenceAction
{
public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_IntroMenuCarsInPosition_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
```

```
};

// Class TAGame.SeqAct_IsActionPressed_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_IsActionPressed_TA : public USequenceAction
{
public:
    class AActor*           Instigator;          // 0x0160 (0x0008)
    [0x0000000000000000]
    struct FName             ActionName;          // 0x0168 (0x0008)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_IsActionPressed_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
};

// Class TAGame.SeqAct_IsCrateRouletteEnabled_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_IsCrateRouletteEnabled_TA : public USequenceAction
{
public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqAct_IsCrateRouletteEnabled_TA");
        }

        return uClassPointer;
    }

    void eventActivated();
};

// Class TAGame.SeqAct_IsHumanPlayer_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_IsHumanPlayer_TA : public USequenceAction
{
```

```
public:
class AActor* [0x0000000000000000] Instigator; // 0x0160 (0x0008)
class AActor* [0x0000000000000000] Target; // 0x0168 (0x0008)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_IsHumanPlayer_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_IsTargetInPlayerCameraView
// 0x0014 (0x0178 - 0x018C)
class USeqAct_IsTargetInPlayerCameraView : public USeqAct_Latent
{
public:
class AActor* [0x0000000000000000] Instigator; // 0x0178 (0x0008)
class AActor* [0x0000000000000000] Target; // 0x0180 (0x0008)
float MaxConeAngle; // 0x0188 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_IsTargetInPlayerCameraView");
}

return uClassPointer;
};

bool eventUpdate(float DeltaTime);

// Class TAGame.SeqAct_LoadMap_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_LoadMap_TA : public USequenceAction
{
```

```
public:
class FString Map; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_LoadMap_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_LookAtActor_TA
// 0x0018 (0x0178 - 0x0190)
class USeqAct_LookAtActor_TA : public USeqAct_Latent
{
public:
class AActor* Actor; // 0x0178 (0x0008)
[0x0000000000000000]
class AActor* ActorToLookAt; // 0x0180 (0x0008)
[0x0000000000000000]
float InterpSpeed; // 0x0188 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bConstantInterpSpeed : 1; // 0x018C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bUpdating : 1; // 0x018C (0x0004)
[0x0000000000000000] [0x00000002]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_LookAtActor_TA");
}

return uClassPointer;
};

void SetActorRotation(struct FRotator NewRotation);
bool eventUpdate(float DeltaTime);
void eventActivated();
};
```

```
// Class TAGame.SeqAct_MainMenuSwitch_TA
// 0x0000 (0x0160 - 0x0160)
class USeqAct_MainMenuSwitch_TA : public USequenceAction
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_MainMenuSwitch_TA");
}

return uClassPointer;
};

static int32_t eventGetObjClassVersion();
void eventActivated();
};

// Class TAGame.SeqAct_MatchCountdown_TA
// 0x0004 (0x0160 - 0x0164)
class USeqAct_MatchCountdown_TA : public USequenceAction
{
public:
int32_t           CountdownTime;           // 0x0160 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_MatchCountdown_TA");
}

return uClassPointer;
};

void HandleCountdown();
void eventActivated();
};

// Class TAGame.SeqAct_MoveActor_TA
// 0x0018 (0x0178 - 0x0190)
class USeqAct_MoveActor_TA : public USeqAct_Latent
{
public:
class AActor*          Actor;           // 0x0178 (0x0008)
```

```
[0x0000000000000000]
class AActor* MoveTransform; // 0x0180 (0x0008)
[0x0000000000000000]
float InterpSpeed; // 0x0188 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bConstantInterpSpeed : 1; // 0x018C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bMove : 1; // 0x018C (0x0004)
[0x0000000000000000] [0x00000002]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_MoveActor_TA");
}

return uClassPointer;
};

bool eventUpdate(float DeltaTime);
void eventActivated();
};

// Class TAGame.SeqAct_NotifyEventOnTutorial_TA
// 0x0018 (0x0160 - 0x0178)
class USeqAct_NotifyEventOnTutorial_TA : public USequenceAction
{
public:
class FString EventName; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class AActor* Instigator; // 0x0170 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_NotifyEventOnTutorial_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_RegisterMapDragon_TA
```

```
// 0x0010 (0x0160 - 0x0170)
class USeqAct_RegisterMapDragon_TA : public USequenceAction
{
public:
class AActor*           PlacedActor;          // 0x0160 (0x0008)
[0x0000000000000000]
int32_t                 TeamIndex;            // 0x0168 (0x0004)
[0x0000000000000000]
unsigned long             bActivated : 1;       // 0x016C (0x0004)
[0x0000000000002000] [0x00000001] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_RegisterMapDragon_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_ReinitPhys_TA
// 0x0008 (0x0160 - 0x0168)
class USeqAct_ReinitPhys_TA : public USequenceAction
{
public:
class AActor*           RBObj;                // 0x0160 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ReinitPhys_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_RotateActor_TA
// 0x001C (0x0178 - 0x0194)
class USeqAct_RotateActor_TA : public USeqAct_Latent
```

```

{
public:
class AActor* Actor; // 0x0178 (0x0008)
[0x0000000000000000]
class AActor* RotateTransform; // 0x0180 (0x0008)
[0x0000000000000000]
float InterpSpeed; // 0x0188 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bConstantInterpSpeed : 1; // 0x018C (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bUpdating : 1; // 0x018C (0x0004)
[0x0000000000000000] [0x00000002]
float RotateTolerance; // 0x0190 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_RotateActor_TA");
}

return uClassPointer;
};

void SetActorRotation(struct FRotator NewRotation);
bool eventUpdate(float DeltaTime);
void eventActivated();
};

// Class TAGame.SeqAct_ScreenFade
// 0x0014 (0x0160 - 0x0174)
class USeqAct_ScreenFade : public USequenceAction
{
public:
class AActor* Instigator; // 0x0160 (0x0008)
[0x0000000000000000]
struct FColor FadeColor; // 0x0168 (0x0004)
[0x0000000000000001] (CPF_Edit)
float FadeTime; // 0x016C (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bFadeAudio : 1; // 0x0170 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ScreenFade");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_ScreenFade_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_ScreenFade_TA : public USequenceAction
{
public:
class AActor* Instigator; // 0x0160 (0x0008)
[0x0000000000000000]
struct FColor FadeColor; // 0x0168 (0x0004)
[0x0000000000000001] (CPF_Edit)
float FadeTime; // 0x016C (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bFadeAudio : 1; // 0x0170 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ScreenFade_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SetBallCamTarget_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_SetBallCamTarget_TA : public USequenceAction
{
public:
class AActor* Target; // 0x0160 (0x0008)
[0x0000000000000000]
class APlayerController_TA* Instigator; // 0x0168 (0x0008)
[0x0000000000000000]
unsigned long bForceBallCamTarget : 1; // 0x0170 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{

```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetBallCamTarget_TA");
}

return uClassPointer;
};

void SetBallCamTarget(unsigned long bEnable);
void eventActivated();
};

// Class TAGame.SeqAct_SetBallMaterial_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_SetBallMaterial_TA : public USequenceAction
{
public:
class ABall_TA*           Ball;           // 0x0160 (0x0008)
[0x0000000000000000]
class UMaterialInterface*   Material;       // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t                    MaterialID;     // 0x0170 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetBallMaterial_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SetBreakoutBallCharge_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_SetBreakoutBallCharge_TA : public USequenceAction
{
public:
class AActor*               BallObj;        // 0x0160 (0x0008)
[0x0000000000000000]
int32_t                     Charge;         // 0x0168 (0x0004)
[0x0000000000000000]
int32_t                     Team;          // 0x016C (0x0004)
[0x0000000000000000]

```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetBreakoutBallCharge_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_SetBreakoutTileDamage_TA
// 0x000C (0x0160 - 0x016C)
class USeqAct_SetBreakoutTileDamage_TA : public USequenceAction
{
public:
class AActor* PlatformObj; // 0x0160 (0x0008)
[0x0000000000000000]
int32_t Damage; // 0x0168 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetBreakoutTileDamage_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_SetCarTransform_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_SetCarTransform_TA : public USequenceAction
{
public:
class AActor* Instigator; // 0x0160 (0x0008)
[0x0000000000000000]
class AActor* Transform; // 0x0168 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetCarTransform_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_SetGreenScreen_TA
// 0x0010 (0x0160 - 0x0170)
class USeqAct_SetGreenScreen_TA : public USequenceAction
{
public:
struct FLinearColor           ColorToUse;           // 0x0160 (0x0010)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetGreenScreen_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_SetMatInstParam_Vector
// 0x001C (0x0160 - 0x017C)
class USeqAct_SetMatInstParam_Vector : public USequenceAction
{
public:
struct FName                 ParamName;           // 0x0160 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AMaterialInstanceActor* MatInstActor;         // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector                VectorValue;        // 0x0170 (0x000C)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetMatInstParam_Vector");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SetMatInstVectorParam_TA
// 0x001C (0x0160 - 0x017C)
class USeqAct_SetMatInstVectorParam_TA : public USequenceAction
{
public:
class UMaterialInstanceConstant* MatInst; // 0x0160 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FName ParamName; // 0x0168 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FVector VectorValue; // 0x0170 (0x000C)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetMatInstVectorParam_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SetMeshMaterialScalarParameter_TA
// 0x0024 (0x0160 - 0x0184)
class USeqAct_SetMeshMaterialScalarParameter_TA : public USequenceAction
{
public:
TArray<class UObject*> MeshList; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
class FString ParamName; // 0x0170 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float ScalarValue; // 0x0180 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame.SeqAct_SetMeshMaterialScalarParameter_TA");
}

return uClassPointer;
};

void __SeqAct_SetMeshMaterialScalarParameter_TA__Activated_0x1(class UObject* Mesh);
void eventActivated();
};

// Class TAGame.SeqAct_SetNameplate_TA
// 0x0018 (0x0160 - 0x0178)
class USeqAct_SetNameplate_TA : public USequenceAction
{
public:
class AActor*           Instigator;          // 0x0160 (0x0008)
[0x0000000000000000]
class FString            LocalizedPlayerName; // 0x0168 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetNameplate_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SetPostProcess_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_SetPostProcess_TA : public USequenceAction
{
public:
TArray<class UPostProcessChain*>      Chains;          // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
unsigned long             bReplaceCurrentChain : 1; // 0x0170 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetPostProcess_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_SetReplayCamera_TA
// 0x002C (0x0160 - 0x018C)
class USeqAct_SetReplayCamera_TA : public USequenceAction
{
public:
class FString FocusActor; // 0x0160 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FVector LocationOffset; // 0x0170 (0x000C)
[0x0000000000000001] (CPF_Edit)
struct FRotator Rotation; // 0x017C (0x000C)
[0x0000000000000001] (CPF_Edit)
float FOV; // 0x0188 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetReplayCamera_TA");
}

return uClassPointer;
};

void eventActivated();

// Class TAGame.SeqAct_SetSkelMATLoadout_TA
// 0x0014 (0x0160 - 0x0174)
class USeqAct_SetSkelMATLoadout_TA : public USequenceAction
{
public:
class ASkeletalMeshActorMAT_Products_TA* ActorMAT; // 0x0160
(0x0008) [0x0000000000000000]
class UObject* ControllerOrPawn; // 0x0168 (0x0008)
[0x0000000000000000]
int32_t LoadoutIndex; // 0x0170 (0x0004)
}

```

[0x0000000000000001] (CPF\_Edit)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetSkelMATLoadout_TA");  
}  
  
return uClassPointer;  
};  
  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_SetStadiumTeamColors_TA  
// 0x0028 (0x0160 - 0x0188)  
class USeqAct_SetStadiumTeamColors_TA : public USequenceAction  
{  
public:  
unsigned long           bSwapTeamColors : 1;           // 0x0160 (0x0004)  
[0x0000000000000001] [0x00000001] (CPF_Edit)  
TArray<struct FLinearColor>   BlueColors;           // 0x0168 (0x0010)  
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)  
TArray<struct FLinearColor>   OrangeColors;         // 0x0178 (0x0010)  
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetStadiumTeamColors_TA");  
}  
  
return uClassPointer;  
};  
  
void SwapTeamArchetypeColors(class ATeam_TA* TeamToChange, TArray<struct FLinearColor> NewColors);  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_SetTutorialGameEvent_TA  
// 0x0010 (0x0160 - 0x0170)  
class USeqAct_SetTutorialGameEvent_TA : public USequenceAction  
{  
public:  
class FString           TutorialType;           // 0x0160 (0x0010)
```

[0x0000000000400001] (CPF\_Edit | CPF\_NeedCtorLink)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetTutorialGameEvent_TA");  
}  
  
return uClassPointer;  
};  
  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_SetTutorialTip_TA  
// 0x0020 (0x0160 - 0x0180)  
class USeqAct_SetTutorialTip_TA : public USequenceAction  
{  
public:  
class FString Tip; // 0x0160 (0x0010)  
[0x0000000000408003] (CPF>Edit | CPF_Const | CPF_Localized | CPF_NeedCtorLink)  
TArray<struct FName> ActionNames; // 0x0170 (0x0010)  
[0x0000000000400001] (CPF>Edit | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetTutorialTip_TA");  
}  
  
return uClassPointer;  
};  
  
class FString GetStringWithMergedActions();  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_SetVehicleInput_TA  
// 0x0020 (0x0160 - 0x0180)  
class USeqAct_SetVehicleInput_TA : public USequenceAction  
{  
public:  
struct FVehicleInputs Input; // 0x0160 (0x0020)  
[0x0000000000000001] (CPF>Edit)  
  
public:
```

```

static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SetVehicleInput_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SpawnActor_TA
// 0x0018 (0x0160 - 0x0178)
class USeqAct_SpawnActor_TA : public USequenceAction
{
public:
class AActor* ActorArchetype; // 0x0160 (0x0008)
[0x0000000000000000]
class AActor* Transform; // 0x0168 (0x0008)
[0x0000000000000000]
class AActor* SpawnedActor; // 0x0170 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SpawnActor_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqAct_SpawnBot_TA
// 0x001C (0x0160 - 0x017C)
class USeqAct_SpawnBot_TA : public USequenceAction
{
public:
class ACar_TA* SpawnedBot; // 0x0160 (0x0008)
[0x0000000000000000]
class AActor* SpawnTransform; // 0x0168 (0x0008)
[0x0000000000000000]
float SkillLevel; // 0x0170 (0x0004)
[0x0000000000000001] (CPF_Edit)

```

```

unsigned long bDisableAllLogic : 1; // 0x0174 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bCollideWithBall : 1; // 0x0174 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long bCollideWithVehicle : 1; // 0x0174 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
int32_t TeamIndex; // 0x0178 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SpawnBot_TA");
}

return uClassPointer;
};

void ChooseTeam(class AAIController_TA* AIC);
void eventActivated();
};

// Class TAGame.SeqAct_SpawnCarColors_TA
// 0x0074 (0x0160 - 0x01D4)
class USeqAct_SpawnCarColors_TA : public USequenceAction
{
public:
struct FSpawnCarColorSet AccentColors; // 0x0160 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FLinearColor AccentTeamColor; // 0x0170 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FSpawnCarColorSet BlueTeamColors; // 0x0180 (0x0010)
[0x0000000000000001] (CPF_Edit)
struct FSpawnCarColorSet OrangeTeamColors; // 0x0190
(0x0010) [0x0000000000000001] (CPF_Edit)
struct FLinearColor TeamAccentColor; // 0x01A0 (0x0010)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_Body_TA* Body; // 0x01B0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UProductAsset_Skin_TA* Skin; // 0x01B8 (0x0008)
[0x0000000000000001] (CPF_Edit)
int32_t CarYaw; // 0x01C0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarOffsetX; // 0x01C4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarOffsetY; // 0x01C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float SetOffsetY; // 0x01CC (0x0004)
[0x0000000000000001] (CPF_Edit)
float BaseOffsetY; // 0x01D0 (0x0004)

```

[0x0000000000000000] (CPF\_Transient)

```
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_SpawnCarColors_TA");  
}  
  
return uClassPointer;  
};  
  
class ACarPreviewActor_TA* SpawnCar(struct FLinearColor TeamColor, struct FLinearColor  
AccentColor, float OffsetX, float OffsetY);  
struct FLinearColor GetRGBFromColorSet(struct FLinearColor Hue, struct FLinearColor SV);  
void SpawnTeamCars(class UCarColorSet_TA* TeamColorSet, struct FLinearColor AccentColor);  
void SpawnAccentCars(class UCarColorSet_TA* AccentColorSet, struct FLinearColor  
TeamColor);  
class UProductAsset_Body_TA* GetOctaneBody();  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_StartTutorial_TA  
// 0x0018 (0x0160 - 0x0178)  
class USeqAct_StartTutorial_TA : public USequenceAction  
{  
public:  
class AActor* Instigator; // 0x0160 (0x0008)  
[0x0000000000000000]  
class FString TutorialName; // 0x0168 (0x0010)  
[0x000080000400003] (CPF_Edit | CPF_Const | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_StartTutorial_TA");  
}  
  
return uClassPointer;  
};  
  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_TermPhys_TA  
// 0x0008 (0x0160 - 0x0168)  
class USeqAct_TermPhys_TA : public USequenceAction
```

```
{  
public:  
class AActor* RBObj; // 0x0160 (0x0008)  
[0x0000000000000000]  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_TermPhys_TA");  
}  
  
return uClassPointer;  
};  
  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_ToggleUIOverlay_TA  
// 0x0010 (0x0160 - 0x0170)  
class USeqAct_ToggleUIOverlay_TA : public USequenceAction  
{  
public:  
class FString UIOverlayName; // 0x0160 (0x0010)  
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_ToggleUIOverlay_TA");  
}  
  
return uClassPointer;  
};  
  
void eventActivated();  
};  
  
// Class TAGame.SeqAct_UpdateAd_TA  
// 0x0008 (0x0160 - 0x0168)  
class USeqAct_UpdateAd_TA : public USequenceAction  
{  
public:  
class AStaticMeshActor* MeshActor; // 0x0160 (0x0008)  
[0x0000000000000000]  
  
public:
```

```
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqAct_UpdateAd_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqCond_CompareString_TA
// 0x0028 (0x0140 - 0x0168)
class USeqCond_CompareString_TA : public USequenceCondition
{
public:
unsigned long          bIgnoreCase : 1;           // 0x0140 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
class FString           StringA;                 // 0x0148 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
class FString           StringB;                 // 0x0158 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqCond_CompareString_TA");
}

return uClassPointer;
};

void eventActivated();
};

// Class TAGame.SeqEvent_CarSpawned_TA
// 0x0014 (0x017C - 0x0190)
class USeqEvent_CarSpawned_TA : public USequenceEvent
{
public:
class FString           PlayerName;            // 0x0180 (0x0010)
[0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_CarSpawned_TA");
}

return uClassPointer;
};

void TriggerFor(class ACar_TA* Car);
void HandleVehicleSetup(class ACar_TA* Car);
void HandleCarSet(class APRI_TA* PRI);
void eventRegisterEvent();
};

// Class TAGame.SeqEvent_CarTouchBall_TA
// 0x0008 (0x017C - 0x0184)
class USeqEvent_CarTouchBall_TA : public USequenceEvent
{
public:
unsigned long           bOneTouchOnly : 1;           // 0x0180 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_CarTouchBall_TA");
}

return uClassPointer;
};

};

// Class TAGame.SeqEvent_IntroMenu_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_IntroMenu_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_IntroMenu_TA");
}
}

```

```

return uClassPointer;
};

};

// Class TAGame.SeqEvent_MainMenuSwitched_TA
// 0x0005 (0x017C - 0x0181)
class USeqEvent_MainMenuSwitched_TA : public USequenceEvent
{
public:
    uint8_t           PrevBackground;          // 0x0180 (0x0001)
    [0x0000000000000000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_MainMenuSwitched_TA");
        }

        return uClassPointer;
    }

    void __SeqEvent_MainMenuSwitched_TA__RegisterEvent_0x2(class UUIConfig_TA* Config);
    void __SeqEvent_MainMenuSwitched_TA__RegisterEvent_0x1(class UUIConfig_TA* Config);
    void HandleMenuBGChange();
    void eventRegisterEvent();
};

// Class TAGame.SeqEvent_StartHighlightReel_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_StartHighlightReel_TA : public USequenceEvent
{
public:

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_StartHighlightReel_TA");
        }

        return uClassPointer;
    }

};

```

```

// Class TAGame.SeqEvent_TutorialStarted_TA
// 0x0004 (0x017C - 0x0180)
class USeqEvent_TutorialStarted_TA : public USequenceEvent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SeqEvent_TutorialStarted_TA");
}

return uClassPointer;
};

};

// Class TAGame.ServerFormPartyEvent_TA
// 0x0008 (0x0060 - 0x0068)
class UServerFormPartyEvent_TA : public UObject
{
public:
class UMergePartyEvent_TA*           ConfirmedMerge;          // 0x0060
(0x0008) [0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerFormPartyEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.ServerMergeErrorEvent_TA
// 0x0068 (0x0060 - 0x00C8)
class UServerMergeErrorEvent_TA : public UObject
{
public:
class UStayAsPartyVote_TA*           Vote;                  // 0x0060 (0x0008)
[0x0000000000000000]
TArray<struct FUniqueNetId>         Invitees;              // 0x0068 (0x0010)
[0x0000000000400000] (CPF_NeedCtorLink)
struct FUniqueNetId                 Leader;                // 0x0078 (0x0048)

```

```
[0x0000000000400000] (CPF_NeedCtorLink)
class UErrorType*           Reason;          // 0x00C0 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerMergeErrorEvent_TA");
}

return uClassPointer;
};

};

// Class TAGame.ServerPerformanceConfig_TA
// 0x0014 (0x0078 - 0x008C)
class UServerPerformanceConfig_TA : public UOnlineConfig_X
{
public:
float                  InternalLowFPS;      // 0x0078 (0x0004)
[0x0000000000000000]
float                  InternalLowFPSTimeThreshold; // 0x007C (0x0004)
[0x0000000000000000]
float                  PublicBadFPS;        // 0x0080 (0x0004)
[0x0000000000000000]
float                  PublicMediocreFPS;    // 0x0084 (0x0004)
[0x0000000000000000]
float                  FpsRollingAverageTime; // 0x0088 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.ServerPerformanceConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.ServerVoteStartedEvent_TA
// 0x0010 (0x0060 - 0x0070)
class UServerVoteStartedEvent_TA : public UObject
{
```

```
public:  
TArray<class AStayAsPartyVoter_TA*> Voters; // 0x0060 (0x0010)  
[0x0000000000400000] (CPF_NeedCtorLink)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ServerVoteStartedEvent_TA");  
}  
  
return uClassPointer;  
};  
};  
  
// Class TAGame.ShopErrors_TA  
// 0x0020 (0x0080 - 0x00A0)  
class UShopErrors_TA : public UErrorList  
{  
public:  
class UErrorType* ShopItemBadData; // 0x0080 (0x0008)  
[0x0001000000000002] (CPF_Const)  
class UErrorType* ShopItemDiscountBadData; // 0x0088 (0x0008)  
[0x0001000000000002] (CPF_Const)  
class UErrorType* ShopSyncedBadData; // 0x0090 (0x0008)  
[0x0001000000000002] (CPF_Const)  
class UErrorType* ShopNotEnoughCurrency; // 0x0098 (0x0008)  
[0x0001000000000002] (CPF_Const)  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.ShopErrors_TA");  
}  
  
return uClassPointer;  
};  
};  
  
// Class TAGame.SkillDecayFeature_TA  
// 0x0000 (0x0060 - 0x0060)  
class USkillDecayFeature_TA : public UObject  
{  
public:
```

```
public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkillDecayFeature_TA");
}

return uClassPointer;
};

// Class TAGame.SkillDecaySystem_TA
// 0x0000 (0x0060 - 0x0060)
class USkillDecaySystem_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SkillDecaySystem_TA");
}

return uClassPointer;
};

static int32_t __SkillDecaySystem_TA__CheckForSkillDecay_0x2(class UGameSettingPlaylist_X* P);
static bool __SkillDecaySystem_TA__CheckForSkillDecay_0x1(class UGameSettingPlaylist_X* P);
static void SetPlaylistSkillSaveData(class UPlaylistSkillDataSave_TA* SkillSave, struct FPlaylistSkillData& Data);
static struct FPlaylistSkillData GetPlaylistSkillData(class UPlaylistSkillDataSave_TA* SkillSave, int32_t Playlist);
static void CheckForSkillDecay(class ULocalPlayer_TA* LocalPlayer, class UPlaylistSkillDataSave_TA* SkillSave, class USyncedSkillData_X* Skill, class UDownloadedPlaylistsData_X* DownloadedPlaylists);
static void HandlePartyJoined(class USkillDecayFeature_TA* Feature, class ULocalPlayer_TA* LocalPlayer, class UPlaylistSkillDataSave_TA* SkillSave, class USyncedSkillData_X* Skill, class UDownloadedPlaylistsData_X* DownloadedPlaylists, class UPartyJoinedEvent_X* PartyJoinedEvent);
static void HandleViewingMatchmaking(class USkillDecayFeature_TA* Feature, class ULocalPlayer_TA* LocalPlayer, class UPlaylistSkillDataSave_TA* SkillSave, class USyncedSkillData_X* Skill, class UDownloadedPlaylistsData_X* DownloadedPlaylists, class UViewingMatchmaking_TA* ViewingMatchmaking);
static void InitSkillDecaySystem(class ULocalPlayer_TA* Player);
};
```

```

// Class TAGame.SkyLightVolume_TA
// 0x000E (0x02A4 - 0x02B2)
class ASkyLightVolume_TA : public AVolume
{
public:
    class USkyLightVolumeComponent_TA*           SkyLight;          // 0x02A8
    (0x0008) [0x0000000004080009] (CPF_Edit | CPF_ExportObject | CPF_Component | CPF_EditInline)
    uint8_t                         UpperPaintTeam;      // 0x02B0 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    uint8_t                         LowerPaintTeam;      // 0x02B1 (0x0001)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.SkyLightVolume_TA");
        }

        return uClassPointer;
    };

    void HandleTeamColorsChanged(class ATeam_TA* Team);
    void HandleAllTeamsCreated(class AGameEvent_Team_TA* TeamGameEvent);
    void HandleGameEventAdded(class AGameEvent_Team_TA* TeamGameEvent);
    void SetTeamColorListeners();
    void eventPostBeginPlay();
};

// Class TAGame.SpawnPointGroup_TA
// 0x0044 (0x0060 - 0x00A4)
class USpawnPointGroup_TA : public UObject
{
public:
    TArray<struct FVector>           SpawnOffset;       // 0x0060 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<struct FRotator>          SpawnFacing;       // 0x0070 (0x0010)
    [0x0000000000400001] (CPF_Edit | CPF_NeedCtorLink)
    TArray<class AActor*>            SpawnPoints;       // 0x0080 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    TArray<class APRI_TA*>           SpotPRI;          // 0x0090 (0x0010)
    [0x0000000000402000] (CPF_Transient | CPF_NeedCtorLink)
    int32_t                          MaxSpots;         // 0x00A0 (0x0004)
    [0x0000000000002000] (CPF_Transient)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpawnPointGroup_TA");
}

return uClassPointer;
};

void RemoveFromSpawnPoints(class AController* ForPlayer);
bool GetSpawnOrientation(class AController* ForPlayer, struct FVector& out_Location, struct
FRotator& out_Rotation);
bool IsPointAvailable(int32_t PointIdx);
int32_t GetPointId(class AActor* Point);
bool TryAddToGroupAtPoint(class AController* Car, int32_t PointIdx);
bool TryAddToGroup(class AController* Car);
void SetupPoints(class AGameEvent_TA* GameEvent, struct FVector& Center, struct FRotator&
Orientation);
};

// Class TAGame.TraceableSpawnPoint_TA
// 0x0008 (0x0268 - 0x0270)
class ATraceableSpawnPoint_TA : public AActor
{
public:
class UCylinderComponent* CylinderComp; // 0x0268 (0x0008)
[0x000000004080008] (CPF_ExportObject | CPF_Component | CPF_EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TraceableSpawnPoint_TA");
}

return uClassPointer;
};

};

// Class TAGame.SpecialEventErrors_TA
// 0x0030 (0x0080 - 0x00B0)
class USpecialEventErrors_TA : public UErrorList
{
public:
class UErrorType* EventNotFound; // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* EventStoreCurrencyExpired; // 0x0088 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType* EventStorePurchaseError; // 0x0090 (0x0008)
[0x0001000000000002] (CPF_Const)

```

```
class UErrorType*           EventStoreNotEnoughCurrency;          // 0x0098
(0x0008) [0x0001000000000002] (CPF_Const)
class UErrorType*           EventStoreLimitReached;             // 0x00A0 (0x0008)
[0x0001000000000002] (CPF_Const)
class UErrorType*           EventStoreDisabled;                // 0x00A8 (0x0008)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialEventErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.SpecialPickup_Attachment_TA
// 0x0010 (0x0388 - 0x0398)
class ASpecialPickup_Attachment_TA : public ASpecialPickup_TA
{
public:
class ASpecialAttachment_TA*      AttachmentArchetype;        // 0x0388
(0x0008) [0x0000000000000001] (CPF_Edit)
class ASpecialAttachment_TA*      Attachment;                 // 0x0390 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Attachment_TA");
}

return uClassPointer;
};

void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_BallCarSpring_TA
// 0x0008 (0x0568 - 0x0570)
class ASpecialPickup_BallCarSpring_TA : public ASpecialPickup_Spring_TA
{
public:
```

```

struct FName SpringMeshDistanceParam; // 0x0568 (0x0008)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BallCarSpring_TA");
}

return uClassPointer;
};

void ScaleSpringMeshToLocation(struct FVector NewLocation, struct FVector TargetLocation);
};

// Class TAGame.SpecialPickup_BallVelcro_TA
// 0x00A8 (0x0388 - 0x0430)
class ASpecialPickup_BallVelcro_TA : public ASpecialPickup_TA
{
public:
struct FVector BallOffset; // 0x0388 (0x000C)
[0x0000000000000001] (CPF_Edit)
unsigned long bUseRealOffset : 1; // 0x0394 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bHit : 1; // 0x0394 (0x0004)
[0x0000008100002020] [0x00000002] (CPF_Net | CPF_Transient)
unsigned long bBroken : 1; // 0x0394 (0x0004)
[0x0000000000002020] [0x00000004] (CPF_Net | CPF_Transient)
float AfterHitDuration; // 0x0398 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UStaticMesh* Mesh; // 0x03A0 (0x0008)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat MeshScaleCurve; // 0x03A8 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat HitMeshScaleCurve; // 0x03C0 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
struct FInterpCurveFloat BrokenScaleCurve; // 0x03D8 (0x0018)
[0x000000000400001] (CPF_Edit | CPF_NeedCtorLink)
float PostBreakDuration; // 0x03F0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue* BallHitSFX; // 0x03F8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float MinBreakForce; // 0x0400 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MinBreakTime; // 0x0404 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CheckLastTouchRate; // 0x0408 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UStaticMeshComponent* SMC; // 0x0410 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)

```

```

class ABall_TA*           WeldedBall;           // 0x0418 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                      OldBallMass;          // 0x0420 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                      AttachTime;          // 0x0424 (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)
float                      LastTouchCheckTime; // 0x0428 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                      BreakTime;           // 0x042C (0x0004)
[0x0000000000002020] (CPF_Net | CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BallVelcro_TA");
}

return uClassPointer;
};

void __SpecialPickup_BallVelcro_TA__HandleHitBall_0x1(class ABall_TA* B);
bool DisableBallImpactForces();
void DoBreak();
void HandleCollision(class ARBActor_TA* RBActor, struct FAccumulatedRigidBodyCollision Collision);
void HandleCarTouch(class ABall_TA* InBall, class ACar_TA* InCar, uint8_t HitType);
void PickupEnd();
void OnBallHit();
void HandleHitBall(class ACar_TA* InCar, class ABall_TA* InBall, struct FVector HitLocation,
struct FVector HitNormal);
void PickupTick(float DeltaTime);
void TickPickupFX(float DeltaTime);
void RemovePickupFX();
void InitPickupFX();
void PickupStart();
void eventReplicatedEvent(struct FName EventName);
};

// Class TAGame.SpecialPickup_Batarang_TA
// 0x0008 (0x0570 - 0x0578)
class ASpecialPickup_Batarang_TA : public ASpecialPickup_BallLasso_TA
{
public:
float                  SpinSpeed;           // 0x0570 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                  CurRotation;         // 0x0574 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Batarang_TA");
}

return uClassPointer;
};

void UpdateVisual(float DeltaTime);

// Class TAGame.SpecialPickup_BoostMod_TA
// 0x000C (0x0388 - 0x0394)
class ASpecialPickup_BoostMod_TA : public ASpecialPickup_TA
{
public:
unsigned long          bUnlimitedBoost : 1;           // 0x0388 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
float                  BoostStrength;             // 0x038C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                  OldBoostStrength;        // 0x0390 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BoostMod_TA");
}

return uClassPointer;
};

void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_BoostOverride_TA
// 0x0038 (0x03C0 - 0x03F8)
class ASpecialPickup_BoostOverride_TA : public ASpecialPickup_Targeted_TA
{
public:
class AFXActor_X*          OtherCarFXArchetype;      // 0x03C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*          BoostSFX;                // 0x03C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UParticleSystem*       BeamPS;                 // 0x03D0 (0x0008)
[0x0000000000000001] (CPF_Edit)
}

```

```
float AddVelocityZ; // 0x03D8 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UParticleSystemComponent* BeamPSC; // 0x03E0
(0x0008) [0x000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class AFXActor_X* OtherCarFX; // 0x03E8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ACar_TA* OtherCar; // 0x03F0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_BoostOverride_TA");
}

return uClassPointer;
};

void PickupEnd();
void RemovePickupFX();
void InitPickupFX();
void OnTargetChanged();
void PickupStart();
};

// Class TAGame.SpecialPickup_CarFreeze_TA
// 0x0020 (0x03C0 - 0x03E0)
class ASpecialPickup_CarFreeze_TA : public ASpecialPickup_Targeted_TA
{
public:
class AFXActor_X* OtherCarFXArchetype; // 0x03C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
float TireFrictionScale; // 0x03C8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AddAngularVelocity; // 0x03CC (0x0004)
[0x0000000000000001] (CPF_Edit)
class AFXActor_X* OtherCarFX; // 0x03D0 (0x0008)
[0x0000000000000000]
class ACar_TA* OtherCar; // 0x03D8 (0x0008)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_CarFreeze_TA");
```

```

}

return uClassPointer;
};

void RemovePickupFX();
void InitPickupFX();
void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_CarGravity_TA
// 0x0014 (0x0388 - 0x039C)
class ASpecialPickup_CarGravity_TA : public ASpecialPickup_TA
{
public:
float GravityScale; // 0x0388 (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FVector AddedForce; // 0x038C (0x000C)
[0x0000000000000001] (CPF_Edit)
float OrigGravityScale; // 0x0398 (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_CarGravity_TA");
}

return uClassPointer;
};

void PickupTick(float DeltaTime);
void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_CarLaunch_TA
// 0x0004 (0x0388 - 0x038C)
class ASpecialPickup_CarLaunch_TA : public ASpecialPickup_TA
{
public:
float LaunchVelocity; // 0x0388 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
}
}

```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_CarLaunch_TA");
}

return uClassPointer;
};

void PickupStart();

// Class TAGame.SpecialPickup_CarSpeed_TA
// 0x0040 (0x0388 - 0x03C8)
class ASpecialPickup_CarSpeed_TA : public ASpecialPickup_TA
{
public:
float SpeedMultiplier; // 0x0388 (0x0004)
[0x0000000000000001] (CPF_Edit)
float MaxLinearSpeedMultiplier; // 0x038C (0x0004)
[0x0000000000000001] (CPF_Edit)
struct FInterpCurveFloat DriveTorqueCurve; // 0x0390 (0x0018)
[0x00000000400001] (CPF_Edit | CPF_NeedCtorLink)
float OriginalTorque; // 0x03A8 (0x0004)
[0x000000000002000] (CPF_Transient)
float OriginalMaxLinearSpeed; // 0x03AC (0x0004)
[0x000000000002000] (CPF_Transient)
struct FInterpCurveFloat OriginalDriveTorqueCurve; // 0x03B0
(0x0018) [0x000000000402000] (CPF_Transient | CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_CarSpeed_TA");
}

return uClassPointer;
};

void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_Demolish_TA
// 0x0004 (0x0388 - 0x038C)
class ASpecialPickup_Demolish_TA : public ASpecialPickup_TA
{
public:
uint8_t DemolishTarget; // 0x0388 (0x0001)
[0x0000000000000001] (CPF_Edit)
uint8_t DemolishSpeed; // 0x0389 (0x0001)

```

```

[0x0000000000000001] (CPF_Edit)
uint8_t OldTarget; // 0x038A (0x0001)
[0x0000000000002000] (CPF_Transient)
uint8_t OldSpeed; // 0x038B (0x0001)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Demolish_TA");
}

return uClassPointer;
};

void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_Football_TA
// 0x0074 (0x0388 - 0x03FC)
class ASpecialPickup_Football_TA : public ASpecialPickup_TA
{
public:
class UAkSoundCue* BallHitSFX; // 0x0388 (0x0008)
[0x0001000000000000]
class UAkSoundCue* DeactivateSFX; // 0x0390 (0x0008)
[0x0001000000000000]
float AttachTime; // 0x0398 (0x0004)
[0x000100000002000] (CPF_Transient)
float DetachTime; // 0x039C (0x0004)
[0x000100000002000] (CPF_Transient)
TArray<struct FThrowSetting> ThrowSettings; // 0x03A0 (0x0010)
[0x000100000400000] (CPF_NeedCtorLink)
struct FVector AttachOffset; // 0x03B0 (0x000C)
[0x0001000000000000]
float CarFumbleForce; // 0x03BC (0x0004)
[0x0001000000000000]
float DoubleJumpUpForce; // 0x03C0 (0x0004)
[0x0001000000000000]
float DemolishUpForce; // 0x03C4 (0x0004)
[0x0001000000000000]
float MinBreakTime; // 0x03C8 (0x0004)
[0x0001000000000000]
float MinAttachTime; // 0x03CC (0x0004)
[0x0001000000000000]
float CheckLastTouchRate; // 0x03D0 (0x0004)
[0x0001000000000000]
float OutOfBoundsZ; // 0x03D4 (0x0004)
[0x0001000000000000]

```

```

class ABall_TA*           WeldedBall;          // 0x03D8 (0x0008)
[0x0001000100002020] (CPF_Net | CPF_Transient)
float                     LastTouchCheckTime; // 0x03E0 (0x0004)
[0x0001000000002000] (CPF_Transient)
class ABall_TA*           PendingBall;         // 0x03E8 (0x0008)
[0x0001000000002000] (CPF_Transient)
struct FVector            PendingBallDodge;    // 0x03F0 (0x000C)
[0x0001000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Football_TA");
}

return uClassPointer;
};

void __SpecialPickup_Football_TA__OnWeldedBallChanged_0x1(class
ACarComponent_Boost_TA* BoostComponent);
bool IsDrivingOutOfBounds();
bool DisableBallImpactForces();
void DeactivationFX();
void HandleBallExplode(class ABall_TA* Ball);
bool GetShouldHideActivateUI();
void DetachBall();
bool GetWasRecentlyAttached();
void HandleCarTouch(class ABall_TA* InBall, class ACar_TA* InCar, uint8_t HitType);
void FumbleBall(float UpwardForce);
void HandleDemolish(class ACar_TA* Victim, struct FDemolishData Data);
void HandleCarDoubleJump(class ACarComponent_DoubleJump_TA* DoubleJumpComponent);
struct FThrowSetting BlendThrowSettings(uint8_t ThrowTypeA, uint8_t ThrowTypeB, float Alpha);
void ApplyThrow(class ABall_TA* Ball, struct FThrowSetting Setting, struct FVector
ThrowDirection);
void ThrowPendingBall();
void HandleCarDodged(class ACarComponent_Dodge_TA* DodgeComponent);
void HandleCollision(class ARBActor_TA* RBActor, struct FAccumulatedRigidBodyCollision
Collision);
void HandleHitBall(class ACar_TA* InCar, class ABall_TA* InBall, struct FVector HitLocation,
struct FVector HitNormal);
void OnWeldedBallChanged();
void PickupTick(float DeltaTime);
void PickupEnd();
void PickupStart();
void eventReplicatedEvent(struct FName VarName);
};

// Class TAGame.SpecialPickup_HandbrakeOverride_TA
// 0x0008 (0x03C0 - 0x03C8)
class ASpecialPickup_HandbrakeOverride_TA : public ASpecialPickup_Targeted_TA

```

```

{
public:
class ACar_TA*           OtherCar;          // 0x03C0 (0x0008)
[0x0000000000000000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_HandbrakeOverride_TA");
}

return uClassPointer;
};

void PickupEnd();
void PickupStart();
};

// Class TAGame.SpecialPickup_HitForce_TA
// 0x0038 (0x0388 - 0x03C0)
class ASpecialPickup_HitForce_TA : public ASpecialPickup_TA
{
public:
unsigned long             bBallForce : 1;          // 0x0388 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long             bCarForce : 1;          // 0x0388 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)
unsigned long             bDemolishCars : 1;        // 0x0388 (0x0004)
[0x0000000000000001] [0x00000004] (CPF_Edit)
float                     BallHitForce;          // 0x038C (0x0004)
[0x0000000000000001] (CPF_Edit)
float                     CarHitForce;          // 0x0390 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AFXActor_TA*        BallHitFXArchetype;    // 0x0398 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_TA*        CarHitFXArchetype;    // 0x03A0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*        HitSFX;              // 0x03A8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                     MinFXTIME;           // 0x03B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                     OrigBallHitForce;     // 0x03B4 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                     OrigCarHitForce;      // 0x03B8 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                     LastFXTIME;          // 0x03BC (0x0004)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_HitForce_TA");
}

return uClassPointer;
};

void PickupEnd();
void HandleCollision(class ARBActor_TA* RB, struct FAccumulatedRigidBodyCollision Collision);
void HandleCollisionFX(class ARBActor_TA* RB, struct FAccumulatedRigidBodyCollision Collision);
void PickupStart();
};

// Class TAGame.SpecialPickup_Inflate_TA
// 0x0010 (0x03C0 - 0x03D0)
class ASpecialPickup_Inflate_TA : public ASpecialPickup_Targeted_TA
{
public:
float ScaleMultiplier; // 0x03C0 (0x0004)
[0x0001000000000001] (CPF_Edit)
float GravityMultiplier; // 0x03C4 (0x0004)
[0x0001000000000001] (CPF_Edit)
class ACar_TA* OtherCar; // 0x03C8 (0x0008)
[0x000100000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Inflate_TA");
}

return uClassPointer;
};

void PickupEnd();
void OnTargetChanged();
void PickupStart();
};

// Class TAGame.SpecialPickup_Rugby_TA
// 0x0060 (0x0388 - 0x03E8)
class ASpecialPickup_Rugby_TA : public ASpecialPickup_TA
{
public:
class UStaticMesh* Mesh; // 0x0388 (0x0008)
}

```

```

[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*                               BallHitSFX;                      // 0x0390 (0x0008)
[0x0000000000000001] (CPF_Edit)
class UAkSoundCue*                               DeactivateSFX;                   // 0x0398 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                                         AttachTime;                     // 0x03A0 (0x0004)
[0x0000000000002000] (CPF_Transient)
float                                         MinDetachTime;                 // 0x03A4 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                         MinBreakTime;                  // 0x03A8 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                         MinBreakForce;                 // 0x03AC (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                         CheckLastTouchRate;            // 0x03B0 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                                         KickOffActivationDelay;        // 0x03B4 (0x0004)
[0x0000000000000001] (CPF_Edit)
class UStaticMeshComponent*                     SMC;                           // 0x03B8 (0x0008)
[0x0000000004082008] (CPF_ExportObject | CPF_Transient | CPF_Component | CPF_EditInline)
class ABall_TA*                                WeldedBall;                    // 0x03C0 (0x0008)
[0x0000000000002000] (CPF_Transient)
float                                         LastTouchCheckTime;           // 0x03C8 (0x0004)
[0x0000000000002000] (CPF_Transient)
unsigned long                                  bBallWelded : 1;              // 0x03CC (0x0004)
[0x0008000000002020] [0x00000001] (CPF_Net | CPF_Transient)
struct FScriptDelegate                         __bBallWelded_ChangeNotify;    // 0x03D0
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

```

```

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Rugby_TA");
}

return uClassPointer;
};

void __SpecialPickup_Rugby_TA__HandleBallWeldedChanged_0x1(class
ACarComponent_Boost_TA* BoostComponent);
void __bBallWelded_ChangeNotifyFunc();
bool DisableBallImpactForces();
void DeactivationFX();
void HandleBallExploded(class ABall_TA* Ball);
bool GetShouldHideActivateUI();
void DoBreak();
bool GetWasRecentlyAttached();
void HandleCarTouch(class ABall_TA* InBall, class ACar_TA* InCar, uint8_t HitType);
void HandleHitBall(class ACar_TA* InCar, class ABall_TA* InBall, struct FVector HitLocation,
struct FVector HitNormal);
void HandleBallWeldedChanged();

```

```

void HandleCollision(class ARBActor_TA* RBActor, struct FAccumulatedRigidBodyCollision
Collision);
void PickupTick(float DeltaTime);
void RemovePickupFX();
void PickupEnd();
void InitPickupFX();
void PickupStart();
bool GetCanDetach();
bool TryActivate(class ARBActor_TA* TargetOverride);
};

// Class TAGame.SpecialPickup_Swapper_TA
// 0x0028 (0x03C0 - 0x03E8)
class ASpecialPickup_Swapper_TA : public ASpecialPickup_Targeted_TA
{
public:
class AFXActor_TA*           OwnCarFX;           // 0x03C0 (0x0008)
[0x0000000000000001] (CPF_Edit)
class AFXActor_TA*           OtherCarFX;         // 0x03C8 (0x0008)
[0x0000000000000001] (CPF_Edit)
float                         AddVelocityZ;        // 0x03D0 (0x0004)
[0x0000000000000001] (CPF_Edit)
class AFXActor_TA*           OtherFX;            // 0x03D8 (0x0008)
[0x0000000000002000] (CPF_Transient)
class ACar_TA*                OtherCar;           // 0x03E0 (0x0008)
[0x0000000000002000] (CPF_Transient)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_Swapper_TA");
}

return uClassPointer;
};

void PickupEnd();
void InitPickupFX();
void OnTargetChanged();
void PickupStart();
};

// Class TAGame.SpecialPickup_TimeBomb_TA
// 0x0020 (0x0388 - 0x03A8)
class ASpecialPickup_TimeBomb_TA : public ASpecialPickup_TA
{
public:
float                         Radius;             // 0x0388 (0x0004)
[0x0000000000000001] (CPF_Edit)
float                         AlmostReadyDuration; // 0x038C (0x0004)

```

```

[0x0000000000000001] (CPF_Edit)
float StartMatSpeed; // 0x0390 (0x0004)
[0x0000000000000001] (CPF_Edit)
float AlmostReadyMatSpeed; // 0x0394 (0x0004)
[0x0000000000000001] (CPF_Edit)
float ImpulseForce; // 0x0398 (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarVerticalForce; // 0x039C (0x0004)
[0x0000000000000001] (CPF_Edit)
float CarTorque; // 0x03A0 (0x0004)
[0x0000000000000001] (CPF_Edit)
unsigned long bDemolish : 1; // 0x03A4 (0x0004)
[0x0000000000000001] [0x00000001] (CPF_Edit)
unsigned long bImpulse : 1; // 0x03A4 (0x0004)
[0x0000000000000001] [0x00000002] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SpecialPickup_TimeBomb_TA");
}

return uClassPointer;
};

void PickupEnd();
void AlmostReady();
void PickupStart();
};

// Class TAGame.SplineCameraTrack_TA
// 0x0000 (0x02D8 - 0x02D8)
class ASplineCameraTrack_TA : public ASplineActor
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SplineCameraTrack_TA");
}

return uClassPointer;
};

class ASplineActor* GetBestNode(struct FVector FromLocation);

```

```

struct FVector GetLocationOnSpline(struct FVector FromLocation, float ExtraDistance);
};

// Class TAGame.SplineFocusTarget_TA
// 0x000C (0x0268 - 0x0274)
class ASplineFocusTarget_TA : public ACustomFocusTarget_TA
{
public:
class ASplineCameraTrack_TA* Spline; // 0x0268 (0x0008)
[0x0000000000000001] (CPF_Edit)
float DistanceAhead; // 0x0270 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.SplineFocusTarget_TA");
}

return uClassPointer;
};

struct FVector GetFocusLocation(class ACar_TA* ForCar);
};

// Class TAGame.StasisExplosionHandler_TA
// 0x0010 (0x0150 - 0x0160)
class UStasisExplosionHandler_TA : public UExplosionHitHandler_TA
{
public:
struct FVector VelocityToApply; // 0x0150 (0x000C)
[0x0000000000000001] (CPF_Edit)
float SlowDownTime; // 0x015C (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StasisExplosionHandler_TA");
}

return uClassPointer;
};

void ApplyImpulse(class ACar_TA* CarHit, struct FVector HitLocation, float DamageScale, struct FContactInformation& ContactInfo);

```

```

};

// Class TAGame.StatFactory_Basketball_TA
// 0x000C (0x0458 - 0x0464)
class AStatFactory_Basketball_TA : public AStatFactory_TA
{
public:
    class UStatEvent_TA*           HoopsSwishGoal;          // 0x0458 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    float                          RedZoneHeightScale;      // 0x0460 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
            uClassPointer = UObject::FindClass("Class TAGame.StatFactory_Basketball_TA");
        }

        return uClassPointer;
    }

    bool IsSwishGoal(class UGoal_TA* Goal, class ABall_TA* Ball, int32_t ScoreIndex);
    void OnGoalScored(class AGameEvent_Soccar_TA* InGameEvent, class ABall_TA* Ball, class UGoal_TA* Goal, int32_t ScoreIndex, int32_t AssistIndex);
    class UGoal_TA* FindShotGoal(class ABall_TA* Ball);
    bool IsInRedZone(class UGoal_TA* Goal, class ABall_TA* Ball);
};

// Class TAGame.StatFactory_Breakout_TA
// 0x0034 (0x0458 - 0x048C)
class AStatFactory_Breakout_TA : public AStatFactory_TA
{
public:
    class UStatEvent_TA*           BreakoutDamage;         // 0x0458 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    class UStatEvent_TA*           BreakoutDamageLarge;     // 0x0460 (0x0008)
    [0x0000000000000001] (CPF_Edit)
    int32_t                        BreakoutSaveDamageThreshold; // 0x0468 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    int32_t                        BreakoutDamageLargeThreshold; // 0x046C (0x0004)
    [0x0000000000000001] (CPF_Edit)
    float                           ShotGoalTimestep;       // 0x0470 (0x0004)
    [0x0000000000000001] (CPF_Edit)
    class AGameEvent_Breakout_TA*   BreakoutEvent;          // 0x0478
    (0x0008) [0x0000000000002000] (CPF_Transient)
    struct FVector                 FieldCenter;           // 0x0480 (0x000C)
    [0x0000000000000001] (CPF_Transient)

public:
    static UClass* StaticClass()
}

```

```

{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatFactory_Breakout_TA");
}

return uClassPointer;
};

float GetHitDistanceToGoal(class ABall_TA* Ball, struct FBallHitInfo Hit, class UGoal_TA* Goal);
class UGoal_TA* CalculateShotGoal(class UGoal_TA* GoalToTest, class ABall_TA* Ball, float
ArriveTime);
class UGoal_TA* FindShotGoal(class ABall_TA* Ball);
void HandleBallAppliedDamage(class ABall_Breakout_TA* Ball, struct FAppliedBreakoutDamage
Damage);
void UnregisterBall(class APawn_X* Ball);
void RegisterBall(class ABall_TA* Ball);
void SetGameEvent(class AGameEvent_Soccar_TA* InGameEvent);
bool IsInRedZone(class UGoal_TA* Goal, class ABall_TA* Ball);
};

// Class TAGame.StatFactory_HauntedBall_TA
// 0x0004 (0x0458 - 0x045C)
class AStatFactory_HauntedBall_TA : public AStatFactory_TA
{
public:
float EpicSaveTime; // 0x0458 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatFactory_HauntedBall_TA");
}

return uClassPointer;
};

bool IsEpicSave(class ABall_TA* Ball, struct FBallHitInfo Hit, class UGoal_TA* Goal);
bool IsBallBehindGoal(class ABall_TA* Ball, class UGoal_TA* Goal);
class UGoal_TA* FindShotGoal(class ABall_TA* Ball);
};

// Class TAGame.StaticMeshActor_TA
// 0x0008 (0x0268 - 0x0270)
class AStaticMeshActor_TA : public AActor
{
public:

```

```
class UStaticMeshComponent*           StaticMeshComponent;          // 0x0268
(0x0008) [0x0000000040A000B] (CPF_Edit | CPF_Const | CPF_ExportObject | CPF_EditConst | 
CPF_Component | CPF>EditInline)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StaticMeshActor_TA");
}

return uClassPointer;
};

};

// Class TAGame.StatusObserver_FTE_TA
// 0x0010 (0x00A0 - 0x00B0)
class UStatusObserver_FTE_TA : public UStatusObserver_X
{
public:
struct FName           LastGroupCompleted;          // 0x00A0 (0x0008)
[0x0001000000000000]
class UFirstTimeExperienceManager_TA*      FTEManager;          // 0x00A8
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusObserver_FTE_TA");
}

return uClassPointer;
};

void DebugPrint();
void UpdateGroupTrigger(class UStatusTrigger_FTEGroupComplete_TA* InTrigger);
void HandleGroupComplete(struct FName InGroupName);
void HandleManagerAdded(class UFirstTimeExperienceManager_TA* InFTEManager);
void HandleFTESaveAdded(class UFirstTimeExperienceSave_TA* InFTESave);
void Init(class ULocalPlayer_X* LP);
};

// Class TAGame.StatusObserver_MenuStack_TA
// 0x0030 (0x00A0 - 0x00D0)
class UStatusObserver_MenuStack_TA : public UStatusObserver_X
{
```

```

public:
struct FName [0x0000000000000000] LastMenuVisited; // 0x00A0 (0x0008)
struct FName [0x0000000000000000] LastButtonTriggered; // 0x00A8 (0x0008)
struct FName [0x0000000000000000] LastCrumbTrailCompleted; // 0x00B0 (0x0008)
struct FName [0x0001000000000000] LastMenuTreeButtonClicked; // 0x00B8 (0x0008)
class UCrumbTrails_TA* [0x0001800000000000] CrumbManager; // 0x00C0 (0x0008)
class UCrumbTrailSave_TA* [0x000100000002000] (CPF_Transient) CrumbSave; // 0x00C8 (0x0008)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusObserver_MenuStack_TA");
}

return uClassPointer;
};

void DebugPrint();
void UpdateCrumbTrigger(class UStatusTrigger_CrumbPathComplete_TA* InTrigger);
void UpdateMenuTreeButtonTrigger(class UStatusTrigger_MenuTreeButton_TA* InTrigger);
void UpdateMenuButtonTrigger(class UStatusTrigger_MenuButton_TA* InTrigger);
void UpdateMenuTrigger(class UStatusTrigger_Menu_TA* InTrigger);
void HandleCrumbComplete(struct FName InCrumbName);
void HandleMenuTreeButtonChange(struct FName TriggeredButton);
void HandleMenuButtonChange(struct FName TriggeredButton);
void HandleMenuChange(struct FName PrevMenu, struct FName TopMenu, unsigned long bClearingStack);
void SetupCrumbSave();
void HandleCrumbSaveAdded(class UCrumbTrailSave_TA* InSave);
void SaveTriggerSetup();
void Init(class ULocalPlayer_X* LP);
};

// Class TAGame.StatusTrigger_Menu_TA
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_Menu_TA : public UStatusTrigger_Named_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

```

```
if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_Menu_TA");
}

return uClassPointer;
};

};

// Class TAGame.StatusTrigger_MenuTreeButton_TA
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_MenuTreeButton_TA : public UStatusTrigger_Named_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_MenuTreeButton_TA");
}

return uClassPointer;
};

};

// Class TAGame.StatusTrigger_CrumbPathComplete_TA
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_CrumbPathComplete_TA : public UStatusTrigger_Named_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_CrumbPathComplete_TA");
}

return uClassPointer;
};

void HandleChange(struct FName InValue);
};

// Class TAGame.StatusTrigger_ProductOwned_TA
```

```
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_ProductOwned_TA : public UStatusTrigger_Product_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_ProductOwned_TA");
}

return uClassPointer;
};

};

// Class TAGame.StatusTrigger_ProductEquipped_TA
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_ProductEquipped_TA : public UStatusTrigger_Product_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_ProductEquipped_TA");
}

return uClassPointer;
};

};

// Class TAGame.StatusObserver_XPLevel_TA
// 0x0008 (0x00A0 - 0x00A8)
class UStatusObserver_XPLevel_TA : public UStatusObserver_X
{
public:
int32_t LastGivenXp; // 0x00A0 (0x0004)
[0x0000000000000000]
int32_t LastGivenLevel; // 0x00A4 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
```

```

static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.StatusObserver_XPLevel_TA");
}

return uClassPointer;
};

void DebugPrint();
void UpdateLevelTrigger(class UStatusTrigger_PlayerLevel_TA* InTrigger);
void HandleLevelUpdated(class UClientXPSave_TA* XPSave, int32_t NewLevel, int32_t OldLevel);
void UpdateXPTrigger(class UStatusTrigger_PlayerXP_TA* InTrigger);
void HandleTriggerXPUpdate(class UClientXPSave_TA* XPSave, int32_t OldXP);
void HandleSaveLoaded(class UClientXPSave_TA* InXPSave);
void Init(class ULocalPlayer_X* LP);
};

// Class TAGame.StatusTrigger_Numerical_TA
// 0x0008 (0x0080 - 0x0088)
class UStatusTrigger_Numerical_TA : public UStatusTrigger_X
{
public:
    uint8_t TCNTType; // 0x0080 (0x0001)
    [0x0000000000000001] (CPF_Edit)
    int32_t ConditionalValue; // 0x0084 (0x0004)
    [0x0000000000000001] (CPF_Edit)

public:
    static UClass* StaticClass()
    {
        static UClass* uClassPointer = nullptr;

        if (!uClassPointer)
        {
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_Numerical_TA");
        }

        return uClassPointer;
    };

    void DebugPrint();
    void HandleChange(int32_t InValue);
};

// Class TAGame.StatusTrigger_PlayerLevel_TA
// 0x0000 (0x0088 - 0x0088)
class UStatusTrigger_PlayerLevel_TA : public UStatusTrigger_Numerical_TA
{
public:

public:
    static UClass* StaticClass()

```

```
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_PlayerLevel_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.StatusTrigger_PlayerXP_TA  
// 0x0000 (0x0088 - 0x0088)  
class UStatusTrigger_PlayerXP_TA : public UStatusTrigger_Numerical_TA  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.StatusTrigger_PlayerXP_TA");  
}  
  
return uClassPointer;  
};  
  
};  
  
// Class TAGame.StayAsPartyMetrics_TA  
// 0x0000 (0x0080 - 0x0080)  
class UStayAsPartyMetrics_TA : public UMetricsGroup_X  
{  
public:  
  
public:  
static UClass* StaticClass()  
{  
static UClass* uClassPointer = nullptr;  
  
if (!uClassPointer)  
{  
uClassPointer = UObject::FindClass("Class TAGame.StayAsPartyMetrics_TA");  
}  
  
return uClassPointer;  
};  
  
static void PartyMergeResults(TArray<struct FPartyMergeResult>& Results);
```

```

static void ProcessPartyUpResults(class UStayAsPartyVote_TA* Vote, class
UGameEventFinishedState_TA* _);
static void ProcessMergeError(class UStayAsPartyVote_TA* Vote, class UErrorType* InErrorType,
TArray<struct FUniqueNetId>& Invitees, struct FUniqueNetId& Leader);
static void OnMergeError(class UStayAsPartyVote_TA* Vote, class UServerMergeErrorEvent_TA*
ErrorEvent);
static void OnMergeAdded(class UStayAsPartyVote_TA* Vote, class UMergePartyEvent_TA*
MergeEvent);
static void PartyUpActive(TArray<struct FUniqueNetId>& Players);
static void OnPlayerVoted(class AStayAsPartyVoter_TA* Voter);
};

// Class TAGame.Team_Freeplay_TA
// 0x0000 (0x04A8 - 0x04A8)
class ATeam_Freeplay_TA : public ATeam_Soccar_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Team_Freeplay_TA");
}

return uClassPointer;
};

void GetColors(struct FLinearColor& OutFontColor, TArray<struct FLinearColor>& OutColors);
};

// Class TAGame.TourBracketGenerator_TA
// 0x0000 (0x0060 - 0x0060)
class UTourBracketGenerator_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourBracketGenerator_TA");
}

return uClassPointer;
};

static void PrintBracketTree(TArray<struct FTourMatchNode>& Tree);

```

```

static struct FTourLeaderboard CreateLeaderboard();
static TArray<struct FTourMatch> NodesToMatches(TArray<struct FTourMatchNode>& Nodes);
static struct FTourMatchNode CreateNode(int32_t MatchID);
static struct FQWORDPair MakeQWORDPair(uint64_t A, uint64_t B);
static TArray<uint64_t> AddByes(TArray<uint64_t> Teams);
static TArray<uint64_t> CreateTeamIDs(int32_t TeamNum);
static TArray<struct FTourMatchNode> BuildBracketTree(int32_t NumTeams);
static TArray<struct FTourPlayer> CreatePlayers(int32_t TeamSize, int32_t PlayerCount);
static TArray<struct FTourTeam> CreateTeams(int32_t TeamCount, int32_t TeamSize);
static struct FTourBracket Generate(uint64_t TourID, int32_t TeamsRegistered, int32_t TeamSize);
};

// Class TAGame.TourSeeder_TA
// 0x0000 (0x0060 - 0x0060)
class UTourSeeder_TA : public UObject
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourSeeder_TA");
}

return uClassPointer;
};

static TArray<uint64_t> GetSeedingForRound(int32_t Rounds);
static TArray<uint64_t> Append(TArray<uint64_t> A, TArray<uint64_t>& B);
static TArray<uint64_t> PairSeeds(TArray<struct FTeamSeeds> Seeds);
static TArray<uint64_t> SeedTeams(TArray<uint64_t> Teams);
};

// Class TAGame.TourEventMatchComplete_TA
// 0x0020 (0x0060 - 0x0080)
class UTourEventMatchComplete_TA : public UObject
{
public:
class AGameEvent_Soccar_TA*           GameEvent;           // 0x0060 (0x0008)
[0x0001000000002000] (CPF_Transient)
struct FScriptDelegate                __EventMatchComplete__Delegate; // 0x0068
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{

```

```

uClassPointer = UObject::FindClass("Class TAGame.TourEventMatchComplete_TA");
}

return uClassPointer;
};

void OnMatchComplete();
void HandleGRISpawned(class AGRI_X* GRI);
void HandleGameStateChanged(class AGameEvent_TA* G);
void HandleGameEventRemoved(class UObject* Obj);
void HandleGameEventAdded(class AGameEvent_Soccar_TA* InGameEvent);
void HandleGameDataSelected(int32_t PlaylistId, int32_t MutatorIndex);
void Deactivate();
void eventConstruct();
void EventMatchComplete();
};

// Class TAGame.TourGameUpdate_Facelt_TA
// 0x0004 (0x0084 - 0x0088)
class UTourGameUpdate_Facelt_TA : public UTourGameUpdate_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourGameUpdate_Facelt_TA");
}

return uClassPointer;
};

class URPC_X* SendUpdateGameRPC(class UTourGameUpdateDispatcher_TA* Dispatcher,
int32_t InGameNum, struct FTourMatchGame InGameData, uint64_t SeriesWinnerID);
};

// Class TAGame.TourServerInfo_AutoTour_TA
// 0x0000 (0x0078 - 0x0078)
class UTourServerInfo_AutoTour_TA : public UTourServerInfo_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TourServerInfo_AutoTour_TA");
}
}

```

```

}

return uClassPointer;
};

class FString GetTieBreakerRules(struct FTourServerSettings& InSettings);
};

// Class TAGame.TradeConfig_TA
// 0x0004 (0x0078 - 0x007C)
class UTradeConfig_TA : public UOnlineConfig_X
{
public:
int32_t MinimumLevelToTrade; // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TradeConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.TradeErrors_TA
// 0x0058 (0x0080 - 0x00D8)
class UTradeErrors_TA : public UErrorList
{
public:
class UErrorType* AttemptedCurrencyGifting; // 0x0080 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* AttemptedCurrencyExchange; // 0x0088
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType* UserTradeBanned; // 0x0090 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* TradeNotAllowed; // 0x0098 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* OtherPlayerTradeNotAllowed; // 0x00A0
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType* OtherPlayerBanned; // 0x00A8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* TradeMinLevelNotReached; // 0x00B0 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType* TradeMinPlayTimeNotReached; // 0x00B8
(0x0008) [0x0000000000000002] (CPF_Const)
class UErrorType* TradeDailyLimitReached; // 0x00C0 (0x0008)
[0x0000000000000002] (CPF_Const)

```

```
class UErrorType*           TradeExceedCurrencyLimit;          // 0x00C8 (0x0008)
[0x0000000000000002] (CPF_Const)
class UErrorType*           TradePlayerNotInParty;           // 0x00D0 (0x0008)
[0x0000000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TradeErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.TradingMetrics_TA
// 0x0000 (0x0080 - 0x0080)
class UTradingMetrics_TA : public UMetricsGroup_X
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TradingMetrics_TA");
}

return uClassPointer;
};

void InvalidProductOffered(struct FUniqueNetId SenderNetId, int32_t ProductID);
};

// Class TAGame.TrainingMetricsConfig_TA
// 0x0004 (0x0078 - 0x007C)
class UTrainingMetricsConfig_TA : public UOnlineConfig_X
{
public:
int32_t                  BatchThreshold;                  // 0x0078 (0x0004)
[0x0000000000000001] (CPF_Edit)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;
```

```

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingMetricsConfig_TA");
}

return uClassPointer;
};

};

// Class TAGame.TrainingProgressFactory_TA
// 0x0020 (0x0060 - 0x0080)
class UTrainingProgressFactory_TA : public UObject
{
public:
class UTrainingProgressSave_TA* ProgressSave; // 0x0060
(0x0008) [0x0000000000002000] (CPF_Transient)
struct FScriptDelegate _ValidateRound__Delegate; // 0x0068
(0x0018) [0x0000000000400000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.TrainingProgressFactory_TA");
}

return uClassPointer;
};

class UTrainingPackProgress_TA* GetOrCreateProgress(class UTrainingEditorData_TA*
TrainingData, struct FScriptDelegate ValidateRoundCallback);
bool ValidateRound(struct FEditorRoundData RoundData);
};

// Class TAGame.Tutorial_Boost_TA
// 0x0000 (0x0248 - 0x0248)
class UTutorial_Boost_TA : public UTutorial_TA
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Tutorial_Boost_TA");
}

```

```

return uClassPointer;
};

void OnActionNameToDetectReleased();
};

// Class TAGame.Tutorial_Dodge_TA
// 0x000C (0x0248 - 0x0254)
class UTutorial_Dodge_TA : public UTutorial_TA
{
public:
unsigned long          bDidDodge : 1;           // 0x0248 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bBallHitGoal : 1;        // 0x0248 (0x0004)
[0x0000000000000000] [0x00000002]
unsigned long          bCheckFailOnLand : 1;     // 0x0248 (0x0004)
[0x0000000000000000] [0x00000004]
int32_t                Dodges;                 // 0x024C (0x0004)
[0x0000000000000000]
int32_t                MaxDodges;              // 0x0250 (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Tutorial_Dodge_TA");
}

return uClassPointer;
};

void CheckBallHitGoal();
void HandleHitGoal(class ABall_TA* Ball, class UGoal_TA* Goal);
void HandleCarTouch(class ABall_TA* Ball, class ACar_TA* HitCar, uint8_t HitType);
void HandleCarLanded(class ACar_TA* Car);
void OnDodged();
void HandleDodgeActivationChanged(class ACarComponent_TA* CarComponent);
void HandleVehicleSetup(class AGameEvent_Tutorial_TA* InGameEvent, class ACar_TA* Car);
};

// Class TAGame.Tutorial_HandBrake_TA
// 0x0018 (0x0248 - 0x0260)
class UTutorial_HandBrake_TA : public UTutorial_TA
{
public:
unsigned long          bCheckHandbrakeDir : 1;    // 0x0248 (0x0004)
[0x0000000000000000] [0x00000001]
unsigned long          bTriggerTouched : 1;       // 0x0248 (0x0004)
[0x0000000000000000] [0x00000002]

```

```
struct FVector HandbrakeStartDir; // 0x024C (0x000C)
[0x0000000000000000] float MinHandbrakeDotAngleForStart; // 0x0258 (0x0004)
[0x0000000000000000] float MinHandbrakeDotAngleForComplete; // 0x025C (0x0004)
[0x0000000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.Tutorial_HandBrake_TA");
}

return uClassPointer;
};

bool HasGoodStartRotation();
void OnActionNameToDetectReleased();
void OnActionNameToDetectPressed();
void CheckActionNameToDetect();
void NotifyEventOnTutorial(class FString EventName);
};

// Class TAGame.UserBugReportComponent_TA
// 0x0010 (0x0060 - 0x0070)
class UUserBugReportComponent_TA : public UUserBugReportComponent_X
{
public:
class UUserBugReportGeneratedData_TA* BulkData; // 0x0060
(0x0008) [0x0001000000000000]
class UUserBugReportSubmissionData_TA* UserData; // 0x0068
(0x0008) [0x0001000000000000]

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportComponent_TA");
}

return uClassPointer;
};

void ClearPendingData();
bool IsReportPending();
void UploadReport(class UPsyNetConnection_X* Connection);
void WaitForConnection(class UUserBugReportGeneratedData_TA* InBulkData);
```

```
void Start(class FString Message);
};

// Class TAGame.UserBugReportErrors_TA
// 0x0008 (0x0080 - 0x0088)
class UUserBugReportErrors_TA : public UErrorList
{
public:
class UErrorType* ReportTooSoon; // 0x0080 (0x0008)
[0x0001000000000002] (CPF_Const)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.UserBugReportErrors_TA");
}

return uClassPointer;
};

};

// Class TAGame.VehiclePartSkeletalMeshComponent_TA
// 0x0000 (0x07B0 - 0x07B0)
class UVehiclePartSkeletalMeshComponent_TA : public USkeletalMeshComponent
{
public:

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class TAGame.VehiclePartSkeletalMeshComponent_TA");
}

return uClassPointer;
};

};

// Class TAGame.__LoadoutValidation_TA__CorrectDLCOwnership_0x1
// 0x0038 (0x0060 - 0x0098)
class U_LoadoutValidation_TA__CorrectDLCOwnership_0x1 : public UObject
{
public:
class APRI_TA* InPRI; // 0x0060 (0x0008)
[0x0000000000000000]
```

```
struct FScriptDelegate          OnLoadoutModified;           // 0x0068 (0x0018)
[0x0000000000040000] (CPF_NeedCtorLink)
struct FScriptDelegate          OnTeamPaintModified;        // 0x0080 (0x0018)
[0x0000000000040000] (CPF_NeedCtorLink)

public:
static UClass* StaticClass()
{
static UClass* uClassPointer = nullptr;

if (!uClassPointer)
{
uClassPointer = UObject::FindClass("Class
TAGame._LoadoutValidation_TA_CorrectDLCOwnership_0x1");
}

return uClassPointer;
};

void __LoadoutValidation_TA_CorrectDLCOwnership_0x1();
};

/*
#
=====
===== #
#
#
=====
===== #
*/
#endif _MSC_VER
#pragma pack(pop)
#endif
```

Removed: 1040

Added: 1075

Generated at <https://www.textcompare.org/> on 05/06/2024, 18:11:48