

Table of Contents

Global

[_ObjectMakeBase](#)
[ActionCardDto](#)
[ActionDataList](#)
[ActionTypeEnum](#)
[AESEncryption](#)
[AlteredStateData](#)
[AlteredStateEnum](#)
[AlteredStatesDataSO](#)
[AlteredView](#)
[AlteredView.AlteredByTurns](#)
[Analytics](#)
[AnimatorListener](#)
[AtlasTextures](#)
[AuthPanel](#)
[AuthPanelType](#)
[AutoDestroyPS](#)
[BackpackController](#)
[BaseMultipliers](#)
[BattleActionBlockCard](#)
[BattleActionBuffDebuff](#)
[BattleActionChangeShinsei](#)
[BattleActionChangeTerrain](#)
[BattleActionDamage](#)
[BattleActionData](#)
[BattleActionHeal](#)
[BattleActionPutAlteredState](#)
[BattleActionReflect](#)
[BattleActionsBase](#)
[BattleActionsBaseTemplate](#)
[BattleActionSkipTurn](#)
[BattleActionStatSwap](#)
[BattleAlteredStateBase](#)

BattleAlteredStateBleeding
BattleAlteredStateDataSO
BattleAlteredStateEvasionChange
BattleAlteredStateIgnited
BattleAlteredStateRoot
BattleNotificationSystem
BattleTerrainBehavioursBase
BattleTerrainBehaviourSnow
BattleTerrainDataSO
BodyStyle
BodyStyle.BodyPartDressable
BoneAnimationTracker
BracketsData
BracketsTournamentManager
CameraPhasesEnum
CameraPointOfInterestEnum
CamerasAvailableEnum
CameraSpawner
ChallengePlayerController
ChangeColor
ChangeGender
ChangelconLeague
CharacterAnimBehaviour
CharacterAPI
CharacterColorSlot
CharacterDatabase
CharacterSelectorController
CharacterStyleInfo
CharacterStyleRelation
CharacterStyleSlot
ChatBadWordsDB
ChatTextBox
CheckStateOfTournamentDto
CheckTournamentInscription
CheckTournamentStateController
ColorIdRelation
ColorSO

ColorsSO
ColorSwapper
CombatFirebasePetitions
Constants
Cooker
Curtain
CriticsCheck
csEffect_03_ArrowFunction
csEffect_03_ObjectMake
csEffect_76_Animation
csEffect_76_CircleSwordMake
csEffect_76_FloatingSwordMake
csEffect_77_HandMove
csEffect_77_ObjectMake
csEffect_77_ObjectMake2
csEffect_79_CircleMake
csEffect_79_WallRiseDown
csEffect_80_SunMove
csEffectScene
csLaser
csLookAt
csMaterial_Change
csMouseOrbit
csMove
csObjectMake
csObjectMake2
csObjectMake3
csObjectMake4
csObjectMake5
csObjectMake6
csObjectMoveDestroy
csObjectMoveDestroy2
csParticleStop
csRiseDown
csRotate
csScaleChange
CustomRampGeneratorEditor

Decolorator
DelayObjectMake
DemoScenesNavigation
DemoToonVFX
DemoToonVFX.SVA
DisableOtherShinsei
DragonPartType
DtoBugData
DtoDetailBugData
DummyMainMenuNavigation
EffectsScene
EGA_EffectSound
EnableDrink
EnableWhenEnable
Extensions
FirebasePetitionType
Followpoints
Footsteps
ForwardMovement
FowardObjectMake
FreeCam
FreeFlyCamera
FreeLookCam
FrontAttack
FrontMover
GameController
HeadMessages
HideAfterSeconds
HS_CameraController
HS_CameraShaker
IBracketsTournament
InteractWithPlayerController
InteractWithShinsei
ITimer
IUiTrigger
JPGGenerator
LeaderboardEntry

LeaderboardEntryResult
LeaderBoardManager
LightFlickerEffect
LobbyChatView
LobbyCompanionPanel
LocalPlayerData
Logout
LookAtAxis
LookPlayer
MatAnimation
MatAnimation.AttachedEdge
MatchConfirmedDto
MatchData
MatchmakingTrigger
MatchState
Material_Change
MaterialOffset
MaterialReskin
MessageView
MessageView.Message
MMConditionAttribute
MMEnumConditionAttribute
MMReadOnlyAttribute
MouseOrbit
MoveToObject
MoveToTag
MultiCircleObjectMake
MultiObjectMake
NewLeaderBoard
NewLeaderBoard.Division
NFTGenerator
NFTsModel
NodeEnumAttribute
OpenUrl
OrderListener
OrderMessage
OrderMessageType

OwnBattleTracker
OwnBattleTracker.PromediumObject
PalletColors
PartEntity
PartEntityModel
ParticleCollisionInstance
PartIdRelation
PartIndex
PartMultipliers
PartRarityModel
PartSelector
PartSlot
PartStat
PartType
PendingVariableNPC
PetInteraction
PlayerDataManager
PlayerIconController
PlayerUI
PlayFabGetLeaderboard
PlayFabGetLeaderboardAroundPlayer
PlayfabManager
PlayFabUpdatePlayerStatistics
PrefabSpawner
PutBackgroundMusic
PutNightModeForVfx
RampGeneratorTDE
RampGeneratorTDE.Mode
RandomAnim
RankRewardDto
RankRewardEntry
RareThing
RarityType
RecolorBehavior
RecolorBtn
ReportBug
RewardEntry

RewardsRankingSystemController
RotateObject
RotateWithDrag
RotatorPillow
ScaleChange
ScrollArrows
SelectableUi
SelectableUIButton
SerializableVector3
SettingsController
Shinsei
ShinseiAnimBehaviour
ShinseiStatsEnum
ShinseiVault
ShouldSerializeContractResolver
ShowHideDebug
ShowPreviewOfSlot
SimpleObjectMake
SimpleVideoPlayer
SineCameraControllerTopDownEffects
SineUIControllerTopDownEffects
SkyChanger
StageBracketData
StatValueType
StopMovementWhenEnable
SwapColorPropertyNameSO
TargetProjectile
TaskManager
tasks
tasks.task
TerrainChangesDataSO
TestExplorer
ThirdPersonController
TimerService
TMPChangeColor
TournamentBracketsData
TournamentCamera

TrackSwitcher
TranslateMove
TransparentBackgroundScreenshotRecorder
TriggerNotificator
TypesActions
TypesOfTerrainEnum
TypeStatsAndMultipliers
UIColorSwapper
UIDisolver
UiEmoji
UiTrigger
UIViewerTrigger
UnityNewtonsoftJsonSerializer
UnlockedCharacterStyleInfo
UpdateTurnData
Vector3Extensions
VendorHeadTracking
VendorSeeArea
VersusPanelController
VFXPositionData
VFXPositionEnum
VFXtester
VFXTypeData
WhiteMageController
WorldPositionateElement
ZoneTrigger
Timba.Characters
 PartVisual
Timba.Games.CharacterFactory
 CharacterBuilder
 CharacterSelector
 CharacterSlot
 CharacterType
 CharacterUtils
 ColorSwapper3D
 PartsStatDatabaseSO
 ShinseiGenerator

Timba.Games.DynamicCamera
CameraPlaneController

Timba.Games.Networking.Testing
NetworkingTestUI

Timba.Games.Recolor
Recolorable

Timba.Games.SacredTails
DisconnectManager
IWallet
WalletController

Timba.Games.SacredTails.BattleModule
UIHelper

Timba.Games.SacredTails.Lobby
PartyManager
PartyManagerLobby
PartyManagerVault
SacredTeleport
ShinseiMovement
ShinseiVaultController

Timba.Games.SacredTails.Lobby.Chat
ChatModule
Message

Timba.Games.SacredTails.LobbyDatabase
CharacterStateEnum
ChatMessagePayload
DeletePayload
ILobbydatabase
ITournamentDatabase
LobbyPlayerBasePayload
PlayfabLobbyDatabase
PlayfabTournamentDatabase
RequestConstants
TypeOfTournamentRequest

Timba.Games.SacredTails.LobbyNetworking
ConnectionResponse
ILobbyNetworkManager
LobbyNetworkingController

- LobbyNetworkingController.DirectMessage
- LobbyNetworkingController.MessageObject
- PlayersStreamData
- Timba.Games.SacredTails.PopupModule
 - IPopupManager
 - PopupManager
 - PopupManager.ButtonType
- Timba.Games.SacredTails.StoreModule
 - StoreController
 - StoreItem
 - StoreItemData
 - StoreView
- Timba.Games.SacredTails.WalletModule
 - WalletView
- Timba.Packages.Games.ChatModule.Controller
 - LobbyChatController
- Timba.Packages.Games.ChatModule.Core
 - ILobbyChatViewable
 - LobbyChatCoreController
- Timba.Packages.Games.ChatModule.Model
 - LobbyChatModel
- Timba.Packages.Games.ChatModule.Structs
 - MessageDto
 - MessagePayload
- Timba.Packages.Games.PlayerControllerModule
 - IInputHandleable
 - InputHandler
 - InputSource
- Timba.Packages.Games.PlayerControllerModule.Core
 - ThirdPersonCoreController
- Timba.Patterns.ServiceLocator
 - IService
 - ServiceLocator
 - ServiceLocatorConfig
 - SingletonMonoBehaviour<T>
 - SingletonMonoBehaviour<T>
- Timba.Recolor

RecolorablePart3D
Timba.SacredTails.Arena
 ActionCard
 ArenaShinseiSelectionController
 ArenaShinseiSelectionUserPanelController
 AttacksAnimation
 BattleGameMode
 BattleShinseiSpawner
 BattleStatisticsCalculator
 BattleStatisticsCalculatorSO
 BattleUIController
 BattleViewerController
 CardDatabase
 CardPreview
 CardUI
 Combat
 CombatPlayer
 CustomAtributes
 GameSceneManager
 ResourceBarValues
 ShinseiPreviewPanelManager
 ShinseiSlot
 ShinseiStats
 TextboxBase
 Turn
 TurnsController
 UserInfo
 VfxInfo
Timba.SacredTails.Arena.ShinseiType
 ShinseiTypeListClass
 ShinseiTypeScriptable
Timba.SacredTails.AudioIntegration
 AudioSettings
 CallStart
 SimpleAudioBtn
 SimpleAudioCallback
 SimpleAudioToggle

- SimpleButtonSound
- SimpleCustomAudio
- WWiseSacred
- Timba.SacredTails.BattleDebugTool
 - ChangeCardsTest
 - DebugShinseiSlot
 - GetShareGroupData
 - GetShareGroupData.CombatPlayerServer
- Timba.SacredTails.CameraModule
 - CameraTriggerChange
 - ViewerCameraChange
- Timba.SacredTails.CardStoreModule
 - Card
 - CardManagementController
 - CardView
 - Deck
- Timba.SacredTails.CharacterStyle
 - BakeColorsInCharacter
 - CharacterRecolor
 - CharacterStyleButton
 - CharacterStyleController
 - CharacterStyleDatabase
 - PartsOfCharacter
 - PlayerMouseRotator
- Timba.SacredTails.ChatModule
 - ChatEmojis
 - ChatEmojis.DiccionarioOfEmojis
 - EmojiButton
 - FilterController
- Timba.SacredTails.Database
 - AuthController
 - DatabaseConnection
 - IDatabase
- Timba.SacredTails.DialogSystem
 - ActionNode
 - ActionNode.Connection
 - AnswerField

- Conversation
- Dialogable
- DialogGraph
- DialogNode
- DialogNode.Connection
- DialogNodeEditor
- DialogUI
- Timba.SacredTails.Interaction
 - SittableMechanic
- Timba.SacredTails.Lobby
 - PlayerCameraFollow
 - ShinseiSpawner
- Timba.SacredTails.Navigation
 - INavigation
 - NavigationManager
- Timba.SacredTails.OptionsModule
 - GameSettingsController
- Timba.SacredTails.Photoboot
 - IconGeneration
 - ShinseiWardrobe
- Timba.SacredTails.TournamentBehavior
 - SearchAndShow
 - SingleBracketPlayerLabelController
 - TournamentBracketsShowController
 - TournamentCreationController
 - TournamentEntry
 - TournamentReadyController
 - TournamentSlot
 - TournamentSlot.JoinTournamentDto
 - TournamentSlot.TournamentPlayerDataDto
- Timba.SacredTails.UiHelpers
 - BackgroundTypeSwapper
 - CalendarController
 - CalendarDateItem
 - ChangeSelectedWithTab
 - EntryController
 - IconContainerSO

IconSet
IUIHelpable
KeepScrollBar
LeaderboardElement
LeaderboardElement.ElementData
PositionCounter
ScrollButton
ScrollViewSystem
SeekMouseIntoScreen
SelectableButtonGroup
SelectableButtonGroup.ButtonList
TypeIconAsigner
UIGroups
UIGroups.UiGroup
WoldPositionateUiElement
WoldPositionUIHandler
Timba.SacredTails.VFXController
VFXDestroyer
VFXInstancer

Namespace Global

Classes

[_ObjectMakeBase](#)

[ActionCardDto](#)

[ActionDataList](#)

[AESEncryption](#)

[AlteredStateData](#)

[AlteredStatesDataSO](#)

[AlteredView](#)

[AlteredView.AlteredByTurns](#)

[Analytics](#)

[AnimatorListener](#)

[AtlasTextures](#)

[AuthPanel](#)

[AutoDestroyPS](#)

[BackpackController](#)

[BaseMultipliers](#)

[BattleActionBlockCard](#)

[BattleActionBuffDebuff](#)

Logic that allow change value of stats in combat

[BattleActionChangeShinsei](#)

Logic of action that allow you change shinsei in combat

[BattleActionChangeTerrain](#)

[BattleActionDamage](#)

Actions that make damage

[BattleActionData](#)

[BattleActionHeal](#)

This class allow grown up the life values

[BattleActionPutAlteredState](#)

This class control the behavior of Altered States

[BattleActionReflect](#)

This class controls a special attack case when shinsei reflect damage

[BattleActionsBase](#)

[BattleActionsBaseTemplate](#)

[BattleActionSkipTurn](#)

This class allow user send a turn withouth data

[BattleActionStatSwap](#)

This class allow swap between two stats

[BattleAlteredStateBase](#)

[BattleAlteredStateBleeding](#)

[BattleAlteredStateDataSO](#)

[BattleAlteredStateEvasionChange](#)

[BattleAlteredStateIgnited](#)

Specific case of Altered State Ignite

[BattleAlteredStateRoot](#)

The specific case of altered state Root

[BattleNotificationSystem](#)

[BattleTerrainBehavioursBase](#)

[BattleTerrainBehaviourSnow](#)

[BattleTerrainDataSO](#)

[BodyStyle](#)

[BodyStyle.BodyPartDressable](#)

[BoneAnimationTracker](#)

[BracketsData](#)

[BracketsTournamentManager](#)

[CameraSpawner](#)

[ChallengePlayerController](#)

[ChangeColor](#)

[ChangeGender](#)

Change between male and female character and update in playfab data

[ChangelconLeague](#)

[CharacterAnimBehaviour](#)

[CharacterAPI](#)

[CharacterColorSlot](#)

UI Element that represents a color for character

[CharacterDatabase](#)

[CharacterSelectorController](#)

[CharacterStyleInfo](#)

[CharacterStyleRelation](#)

[CharacterStyleSlot](#)

[ChatTextBox](#)

Chat behavior, send messages and show in a UI Elements

[CheckStateOfTournamentDto](#)

[CheckTournamentInscription](#)

Allow hide all objects innecessary in tournament mode, only visual results

[CheckTournamentStateController](#)

[ColorIdRelation](#)

[ColorSO](#)

[ColorsSO](#)

[ColorSwapper](#)

[CombatFirebasePetitions](#)

Deprecate battle using firebase

[Constants](#)

[Cooker](#)

This script controls the behavior of npc cooker in the bar

[Curtain](#)

A component that make a black transition

[csEffect_03_ArrowFunction](#)

[csEffect_03_ObjectMake](#)

[csEffect_76_Animation](#)

[csEffect_76_CircleSwordMake](#)

[csEffect_76_FloatingSwordMake](#)

[csEffect_77_HandMove](#)

[csEffect_77_ObjectMake](#)

[csEffect_77_ObjectMake2](#)

[csEffect_79_CircleMake](#)

[csEffect_79_WallRiseDown](#)

[csEffect_80_SunMove](#)

[csEffectScene](#)

[csLaser](#)

[csLookAt](#)

[csMaterial_Change](#)

[csMouseOrbit](#)

[csMove](#)

[csObjectMake](#)

[csObjectMake2](#)

[csObjectMake3](#)

[csObjectMake4](#)

[csObjectMake5](#)

[csObjectMake6](#)

[csObjectMoveDestroy](#)

[csObjectMoveDestroy2](#)

[csParticleStop](#)

[csRiseDown](#)

[csRotate](#)

[csScaleChange](#)

[CustomRampGeneratorEditor](#)

[Decolorator](#)

[DelayObjectMake](#)

[DemoScenesNavigation](#)

[DemoToonVFX](#)

[DisableOtherShinsei](#)

[DummyMainMenuNavigation](#)

[EffectsScene](#)

[EGA_EffectSound](#)

[EnableDrink](#)

[EnableWhenEnable](#)

[Extensions](#)

[Followpoints](#)

[Footsteps](#)

[Simple behavior that play music according to the animation](#)

[ForwardMovement](#)

[FowardObjectMake](#)

[FreeCam](#)

A simple free camera to be added to a Unity game object.

Keys: wasd / arrows - movement q/e - up/down (local space) r/f - up/down (world space) pageup/pagedown - up/down (world space) hold shift - enable fast movement mode right mouse - enable free look mouse - free look / rotation

[FreeFlyCamera](#)

[FreeLookCam](#)

[FrontAttack](#)

[FrontMover](#)

[GameController](#)

[HeadMessages](#)

[HideAfterSeconds](#)

[HS_CameraController](#)

[HS_CameraShaker](#)

[InteractWithPlayerController](#)

[InteractWithShinsei](#)

[IUiTrigger](#)

[JPGGenerator](#)

[LeaderboardEntry](#)

[LeaderboardEntryResult](#)

[LeaderBoardManager](#)

[LightFlickerEffect](#)

Component which will flicker a linked light while active by changing its intensity between the min and max values given. The flickering can be sharp or smoothed depending on the value of the smoothing parameter.

Just activate / deactivate this component as usual to pause / resume flicker

[LobbyChatView](#)

[LobbyCompanionPanel](#)

[LocalPlayerData](#)

[Logout](#)

[LookAtAxis](#)

[LookPlayer](#)

[MatchConfirmedDto](#)

[MatchmakingTrigger](#)

[MatchState](#)

[Material_Change](#)

[MaterialOffset](#)

[MaterialReskin](#)

[MessageView](#)

[MessageView.Message](#)

[MMConditionAttribute](#)

[MMEnumConditionAttribute](#)

[MMReadOnlyAttribute](#)

[MouseOrbit](#)

[MoveToObject](#)

[MoveToTag](#)

[MultiCircleObjectMake](#)

[MultiObjectMake](#)

[NewLeaderBoard](#)

[NFTGenerator](#)

[NFTsModel](#)

[NodeEnumAttribute](#)

Draw enums correctly within nodes. Without it, enums show up at the wrong positions.

[OpenUrl](#)

[OrderListener](#)

Deprecate system for listen orders, avoiding put a server.

[OwnBattleTracker](#)

[OwnBattleTracker.PromediumObject](#)

[PalletColors](#)

[PartEntity](#)

[PartEntityModel](#)

[ParticleCollisionInstance](#)

[PartIdRelation](#)

[PartIndex](#)

Index of all the current parts of the character

[PartMultipliers](#)

[PartRarityModel](#)

[PartSelector](#)

[PartSlot](#)

[PartStat](#)

[PendingVariableNPC](#)

Call all necessary actions to leave the tournament

[PetInteraction](#)

[PlayerDataManager](#)

[PlayerIconController](#)

[PlayerUI](#)

[PlayFabGetLeaderboard](#)

[PlayFabGetLeaderboardAroundPlayer](#)

[PlayfabManager](#)

This works as bridge with playfab, all client petitions pass first here

[PlayFabUpdatePlayerStatistics](#)

[PrefabSpawner](#)

[PutBackgroundMusic](#)

[PutNightModeForVfx](#)

[RampGeneratorTDE](#)

[RandomAnim](#)

[RankRewardDto](#)

[RankRewardEntry](#)

[RareThing](#)

This script solves a problem with Wwise when you put the game in second plane at return all songs in queue play at time, and control the bird sounds

[RecolorBehavior](#)

[RecolorBtn](#)

[ReportBug](#)

[RewardEntry](#)

[RewardsRankingSystemController](#)

[RotateObject](#)

[RotateWithDrag](#)

[RotatorPillow](#)

[ScaleChange](#)

[ScrollArrows](#)

[SelectableUi](#)

[SelectableUiButton](#)

[SerializableVector3](#)

[SettingsController](#)

[Shinsei](#)

[ShinseiAnimBehaviour](#)

[ShinseiVault](#)

[ShouldSerializeContractResolver](#)

[ShowHideDebug](#)

[ShowPreviewOfSlot](#)

[SimpleObjectMake](#)

[SimpleVideoPlayer](#)

[SineCameraControllerTopDownEffects](#)

[SineUIControllerTopDownEffects](#)

[SkyChanger](#)

[StageBracketData](#)

[StopMovementWhenEnable](#)

[SwapColorPropertyNameSO](#)

TargetProjectile

TaskManager

tasks

tasks.task

TerrainChangesDataSO

TestExplorer

ThirdPersonController

Controll the behavior of the main character

TimerService

TMPChangeColor

TournamentBracketsData

TournamentCamera

TrackSwitcher

TranslateMove

TransparentBackgroundScreenshotRecorder

TriggerNotificator

TypesActions

TypeStatsAndMultipliers

UIColorSwapper

UIDisolver

UiEmoji

UiTrigger

UIViewerTrigger

UnityNewtonsoftJsonSerializer

UnlockedCharacterStyleInfo

Vector3Extensions

VendorHeadTracking

VendorSeeArea

VersusPanelController

VFXPositionData

VFXtester

VFXTypeData

WhiteMageController

WorldPositionateElement

ZoneTrigger

Structs

ChatBadWordsDB

DemoToonVFX.SVA

DtoBugData

DtoDetailBugData

MatAnimation

MatchData

OrderMessage

UpdateTurnData

Interfaces

IBracketsTournament

ITimer

Enums

ActionTypeEnum

AlteredStateEnum

AuthPanelType

CameraPhasesEnum

CameraPointOfInterestEnum

CamerasAvailableEnum

CriticsCheck

DragonPartType

FirebasePetitionType

MatAnimation.AttachedEdge

NewLeaderBoard.Division

OrderMessageType

PartType

RampGeneratorTDE.Mode

RarityType

ShinseiStatsEnum

StatValueType

Stat value categories, used of stat calculation

TypesOfTerrainEnum

VFXPositionEnum

Class _ObjectMakeBase

Inheritance

- System.Object
- _ObjectMakeBase
- [DelayObjectMake](#)
- [FowardObjectMake](#)
- [MultiCircleObjectMake](#)
- [MultiObjectMake](#)
- [SimpleObjectMake](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class _ObjectMakeBase : MonoBehaviour
```

Fields

m_makeObjs

Declaration

```
public GameObject[] m_makeObjs
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

m_movePos

Declaration

```
public Transform m_movePos
```

Field Value

TYPE	DESCRIPTION
Transform	

Methods

GetRandomValue(Single)

Declaration

```
public float GetRandomValue(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	

Returns

TYPE	DESCRIPTION
System.Single	

GetRandomVector(Vector3)

Declaration

```
public Vector3 GetRandomVector(Vector3 value)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	value	

Returns

TYPE	DESCRIPTION
Vector3	

Class ActionCardDto

Inheritance

System.Object
ActionCardDto

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class ActionCardDto : ActionCard
```

Fields

indexCard

Declaration

```
public int indexCard
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class ActionDataList

Inheritance

System.Object
ActionDataList

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class ActionDataList : SerializedScriptableObject
```

Fields

actions

Declaration

```
public List<BattleActionData> actions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionData >	

Enum ActionTypeEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public enum ActionTypeEnum
```

Fields

NAME	DESCRIPTION
BlockAction	
BuffDebuff	
ChangeShinsei	
CopyCat	
Damage	
Healing	
PutAlteredState	
Randomize	
ReflectDamage	
SkipTurn	
StatSwap	
TerrainChange	

Class AESEncryption

Inheritance

System.Object

AESEncryption

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class AESEncryption
```

Fields

IV

Declaration

```
public const string IV = "ABCDEFGHJKLMNOP"
```

Field Value

TYPE	DESCRIPTION
System.String	

Key

Declaration

```
public const string Key = "ABCDEFGHJKLMNOPQRSTUVWXYZABCDEFG"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

Decrypt(String)

Declaration

```
public static string Decrypt(string encryptedText)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	encryptedText	

Returns

TYPE	DESCRIPTION
System.String	

Encrypt(String)

Declaration

```
public static string Encrypt(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

Returns

TYPE	DESCRIPTION
System.String	

Class AlteredStateData

Inheritance

System.Object

AlteredStateData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class AlteredStateData
```

Fields

amount

Declaration

```
public int amount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

isTargetLocalPlayer

Declaration

```
public bool isTargetLocalPlayer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

perTurns

Declaration

```
public bool perTurns
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

turnsDuration

Declaration

```
public int turnsDuration
```

Field Value

TYPE	DESCRIPTION
System.Int32	

turnsLeft

Declaration

public int turnsLeft

Field Value

TYPE	DESCRIPTION
System.Int32	

Enum AlteredStateEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum AlteredStateEnum
```

Fields

NAME	DESCRIPTION
Bleeding	
EvasionChange	
Ignited	
Rooted	

Class AlteredStatesDataSO

Inheritance

System.Object

AlteredStatesDataSO

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class AlteredStatesDataSO : SerializedScriptableObject
```

Fields

alteredStates

Declaration

```
public List<BattleAlteredStateDataSO> alteredStates
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleAlteredStateDataSO >	

Methods

GetJsonAlteredStates()

Declaration

```
public void GetJsonAlteredStates()
```

Class AlteredView

Inheritance

System.Object

AlteredView

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AlteredView : MonoBehaviour
```

Fields

spriteTest

Declaration

```
public Sprite spriteTest
```

Field Value

TYPE	DESCRIPTION
Sprite	

testTurn

Declaration

```
public int testTurn
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

PassTurn()

Declaration

```
public void PassTurn()
```

ShowAlteredByTime(Int32, Sprite)

Declaration

```
public void ShowAlteredByTime(int turnCount, Sprite icon)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	turnCount	
Sprite	icon	

Test()

Declaration

```
public void Test()
```

UpdateAlteredVisual(GameObject, String, Sprite)

Declaration

```
public void UpdateAlteredVisual(GameObject target, string text, Sprite icon = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	target	
System.String	text	
Sprite	icon	

Class AlteredView.AlteredByTurns

Inheritance

System.Object

AlteredView.AlteredByTurns

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class AlteredByTurns
```

Fields

alteredObject

Declaration

```
public GameObject alteredObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

turns

Declaration

```
public int turns
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class Analytics

Inheritance

System.Object

Analytics

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Analytics : MonoBehaviour
```

Methods

UpdateAnalitic(String, Int32)

Declaration

```
public void UpdateAnalitic(string statistic, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	statistic	
System.Int32	value	

UpdateAnaliticCounter(String)

Declaration

```
public void UpdateAnaliticCounter(string statistic)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	statistic	

Class AnimatorListener

Inheritance

System.Object

AnimatorListener

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AnimatorListener : NetworkBehaviour
```

Class AtlasTextures

Inheritance

System.Object

AtlasTextures

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class AtlasTextures
```

Fields

AmbientOcclusion

Declaration

```
public Texture2D AmbientOcclusion
```

Field Value

TYPE	DESCRIPTION
Texture2D	

Difuse

Declaration

```
public Texture2D Difuse
```

Field Value

TYPE	DESCRIPTION
Texture2D	

Metallic

Declaration

```
public Texture2D Metallic
```

Field Value

TYPE	DESCRIPTION
Texture2D	

Normal

Declaration

```
public Texture2D Normal
```

Field Value

TYPE	DESCRIPTION
Texture2D	

Class AuthPanel

Inheritance

System.Object
AuthPanel

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class AuthPanel
```

Fields

authPanelType

Declaration

```
public List<AuthPanelType> authPanelType
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< AuthPanelType >	

panelItem

Declaration

```
public Transform panelItem
```

Field Value

TYPE	DESCRIPTION
Transform	

Enum AuthPanelType

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum AuthPanelType
```

Fields

NAME	DESCRIPTION
FORGOT_PASSWORD	
LOGIN	
SIGN_UP	

Class AutoDestroyPS

Inheritance

System.Object

AutoDestroyPS

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AutoDestroyPS : MonoBehaviour
```

Class BackpackController

Inheritance

System.Object

BackpackController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BackpackController : MonoBehaviour
```

Fields

backpackPanel

Declaration

```
public GameObject backpackPanel
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

CloseBackPack()

Declaration

```
public void CloseBackPack()
```

OpenBackPack()

Declaration

```
public void OpenBackPack()
```

Class BaseMultipliers

Inheritance

System.Object

BaseMultipliers

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BaseMultipliers
```

Fields

statBonus1

Declaration

```
public ShinseiStatsEnum statBonus1
```

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

statBonus2

Declaration

```
public ShinseiStatsEnum statBonus2
```

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

statPenalty

Declaration

```
public ShinseiStatsEnum statPenalty
```

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

Class BattleActionBlockCard

Inheritance

System.Object
BattleActionsBase
BattleActionBlockCard

Inherited Members

- BattleActionsBase.actionType
- BattleActionsBase.camManager
- BattleActionsBase.vFXInstancer
- BattleActionsBase.battleGameMode
- BattleActionsBase.battleUIController
- BattleActionsBase.shinseiTypeScriptable
- BattleActionsBase.launchVfx
- BattleActionsBase.vfxTime
- BattleActionsBase.actionTime
- BattleActionsBase.targetName
- BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)
- BattleActionsBase.CheckAction(Single, Action)
- BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)
- BattleActionsBase.ShowBothShinseisFar()
- BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)
- BattleActionsBase.ActionTime()

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionBlockCard : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	battleActionData	

Overrides

[BattleActionsBase.EndAction\(Boolean, UserInfo, UserInfo, BattleActionData\)](#)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData battleActionData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	battleActionData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionBuffDebuff

Logic that allow change value of stats in combat

Inheritance

System.Object

BattleActionsBase

BattleActionBuffDebuff

Inherited Members

BattleActionsBase.actionType

BattleActionsBase.camManager

BattleActionsBase.vFXInstancer

BattleActionsBase.battleGameMode

BattleActionsBase.battleUIController

BattleActionsBase.shinseiTypeScriptable

BattleActionsBase.launchVfx

BattleActionsBase.vfxTime

BattleActionsBase.actionTime

BattleActionsBase.targetName

BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)

BattleActionsBase.CheckAction(Single, Action)

BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)

BattleActionsBase.ShowBothShinseisFar()

BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)

BattleActionsBase.ActionTime()

Namespace: Global

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionBuffDebuff : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData buffDebuffData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	buffDebuffData	

Overrides

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData buffDebuffData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	buffDebuffData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionChangeShinsei

Logic of action that allow you change shinsei in combat

Inheritance

System.Object

[BattleActionsBase](#)

BattleActionChangeShinsei

Inherited Members

[BattleActionsBase.actionType](#)

[BattleActionsBase.camManager](#)

[BattleActionsBase.vFXInstancer](#)

[BattleActionsBase.battleGameMode](#)

[BattleActionsBase.battleUIController](#)

[BattleActionsBase.shinseiTypeScriptable](#)

[BattleActionsBase.launchVfx](#)

[BattleActionsBase.vfxTime](#)

[BattleActionsBase.actionTime](#)

[BattleActionsBase.targetName](#)

[BattleActionsBase.Init\(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable\)](#)

[BattleActionsBase.CheckAction\(Single, Action\)](#)

[BattleActionsBase.DecideIndexes\(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData\)](#)

[BattleActionsBase.ShowBothShinseisFar\(\)](#)

[BattleActionsBase.PlayAnimation\(UserInfo, AttacksAnimation, Boolean, BattleActionData\)](#)

Namespace: [Global](#)

Assembly: [cs.temp.dll.dll](#)

Syntax

```
public class BattleActionChangeShinsei : BattleActionsBase
```

Methods

ActionTime()

Declaration

```
public override float ActionTime()
```

Returns

TYPE	DESCRIPTION
System.Single	

Overrides

[BattleActionsBase.ActionTime\(\)](#)

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	battleActionData	

Overrides

[BattleActionsBase.EndAction\(Boolean, UserInfo, UserInfo, BattleActionData\)](#)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData battleActionData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	battleActionData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionChangeTerrain

Inheritance

System.Object
BattleActionsBase
BattleActionChangeTerrain

Inherited Members

- BattleActionsBase.actionType
- BattleActionsBase.camManager
- BattleActionsBase.vFXInstancer
- BattleActionsBase.battleGameMode
- BattleActionsBase.battleUIController
- BattleActionsBase.shinseiTypeScriptable
- BattleActionsBase.launchVfx
- BattleActionsBase.vfxTime
- BattleActionsBase.actionTime
- BattleActionsBase.targetName
- BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)
- BattleActionsBase.CheckAction(Single, Action)
- BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)
- BattleActionsBase.ShowBothShinseisFar()
- BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)
- BattleActionsBase.ActionTime()

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionChangeTerrain : BattleActionsBase
```

Fields

terrainsBehaviours

Declaration

```
public List<BattleTerrainBehavioursBase> terrainsBehaviours
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleTerrainBehavioursBase >	

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData changeTerrainData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	changeTerrainData	

Overrides

[BattleActionsBase.EndAction\(Boolean, UserInfo, UserInfo, BattleActionData \)](#)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction( bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData changeTerrainData, Action onEndVfxCallback = null )
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	changeTerrainData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action \)](#)

Class BattleActionDamage

Actions that make damage

Inheritance

System.Object

BattleActionsBase

BattleActionDamage

Inherited Members

BattleActionsBase.actionType

BattleActionsBase.camManager

BattleActionsBase.vFXInstancer

BattleActionsBase.battleGameMode

BattleActionsBase.battleUIController

BattleActionsBase.shinseiTypeScriptable

BattleActionsBase.launchVfx

BattleActionsBase.vfxTime

BattleActionsBase.actionTime

BattleActionsBase.targetName

BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)

BattleActionsBase.CheckAction(Single, Action)

BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)

BattleActionsBase.ShowBothShinseisFar()

BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)

BattleActionsBase.ActionTime()

Namespace: Global

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionDamage : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	battleActionData	

Overrides

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData battleActionData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	battleActionData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionData

Inheritance

System.Object

BattleActionData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class BattleActionData : ICloneable
```

Fields

actionElementType

Declaration

```
public CharacterType actionElementType
```

Field Value

TYPE	DESCRIPTION
CharacterType	

actionType

Declaration

```
public ActionTypeEnum actionType
```

Field Value

TYPE	DESCRIPTION
ActionTypeEnum	

activateAlteredState

Declaration

```
public bool activateAlteredState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

alteredState

Declaration

```
public AlteredStateEnum alteredState
```

Field Value

TYPE	DESCRIPTION
AlteredStateEnum	

alteredStateToActivate

Declaration

public AlteredStateEnum alteredStateToActivate
--

Field Value

TYPE	DESCRIPTION
AlteredStateEnum	

amount

Declaration

public int amount

Field Value

TYPE	DESCRIPTION
System.Int32	

applyEachTurn

Declaration

public bool applyEachTurn

Field Value

TYPE	DESCRIPTION
System.Boolean	

bonusPercent

Declaration

public float bonusPercent

Field Value

TYPE	DESCRIPTION
System.Single	

cardSkipTurn

Declaration

public bool cardSkipTurn

Field Value

TYPE	DESCRIPTION
System.Boolean	

casterAnim

Declaration

[NonSerialized] public AttacksAnimation casterAnim

Field Value

TYPE	DESCRIPTION
AttacksAnimation	

changeMinAndMaxStats

Declaration

public bool changeMinAndMaxStats

Field Value

TYPE	DESCRIPTION
System.Boolean	

criticsPercentChange

Declaration

public int criticsPercentChange

Field Value

TYPE	DESCRIPTION
System.Int32	

criticsRoll

Declaration

public float criticsRoll

Field Value

TYPE	DESCRIPTION
System.Single	

currentVFXPositions

Declaration

[NonSerialized] public Dictionary<VFXPositionEnum, Transform> currentVFXPositions
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary< VFXPositionEnum , Transform>	

evadedTurns

Declaration

public int evadedTurns

Field Value

TYPE	DESCRIPTION
System.Int32	

evadeRoll

Declaration

public float evadeRoll

Field Value

TYPE	DESCRIPTION
System.Single	

isBuff

Declaration

public bool isBuff

Field Value

TYPE	DESCRIPTION
System.Boolean	

isComingFromCopyIndex

Declaration

[NonSerialized] public int isComingFromCopyIndex

Field Value

TYPE	DESCRIPTION
System.Int32	

isPercertange

Declaration

public bool isPercertange

Field Value

TYPE	DESCRIPTION
System.Boolean	

isSelfInflicted

Declaration

public bool isSelfInflicted

Field Value

TYPE	DESCRIPTION
System.Boolean	

isVfxReversed

Declaration

[NonSerialized] public bool isVfxReversed
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

launchVfx

Declaration

[NonSerialized] public bool launchVfx
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

numberOfTimesBuffApplied

Declaration

public int numberOfTimesBuffApplied

Field Value

TYPE	DESCRIPTION
System.Int32	

perTurns

Declaration

public bool perTurns

Field Value

TYPE	DESCRIPTION
System.Boolean	

stat1

Declaration

public ShinseiStatsEnum stat1

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

stat2

Declaration

public ShinseiStatsEnum stat2

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

statBonusDamage

Declaration

public ShinseiStatsEnum statBonusDamage

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

statToModify

Declaration

public ShinseiStatsEnum statToModify

Field Value

TYPE	DESCRIPTION
ShinseiStatsEnum	

targetAnim

Declaration

[NonSerialized] public AttacksAnimation targetAnim

Field Value

TYPE	DESCRIPTION
AttacksAnimation	

turnActions

Declaration

[NonSerialized] public ActionCardDto turnActions

Field Value

TYPE	DESCRIPTION
ActionCardDto	

turnsDuration

Declaration

public int turnsDuration

Field Value

TYPE	DESCRIPTION
System.Int32	

turnsPassed

Declaration

public int turnsPassed

Field Value

TYPE	DESCRIPTION
System.Int32	

typeOfTerrain

Declaration

public TypesOfTerrainEnum typeOfTerrain

Field Value

TYPE	DESCRIPTION
TypesOfTerrainEnum	

vfxAffectBoth

Declaration

[NonSerialized] public bool vfxAffectBoth
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

vfxIndex

Declaration

[NonSerialized] public int vfxIndex
--

Field Value

TYPE	DESCRIPTION
System.Int32	

vfxTime

Declaration

[NonSerialized] public float vfxTime

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

Clone()

Declaration

public object Clone()

Returns

TYPE	DESCRIPTION
System.Object	

Class BattleActionHeal

This class allow grown up the life values

Inheritance

System.Object
BattleActionsBase
BattleActionHeal

Inherited Members

BattleActionsBase.actionType
BattleActionsBase.camManager
BattleActionsBase.vFXInstancer
BattleActionsBase.battleGameMode
BattleActionsBase.battleUIController
BattleActionsBase.shinseiTypeScriptable
BattleActionsBase.launchVfx
BattleActionsBase.vfxTime
BattleActionsBase.actionTime
BattleActionsBase.targetName
BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)
BattleActionsBase.CheckAction(Single, Action)
BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)
BattleActionsBase.ShowBothShinseisFar()
BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)
BattleActionsBase.ActionTime()

Namespace: Global
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionHeal : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	battleActionData	

Overrides

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData healData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	healData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionPutAlteredState

This class control the behavior of Altered States

Inheritance

System.Object
BattleActionsBase
BattleActionPutAlteredState

Inherited Members

BattleActionsBase.actionType
BattleActionsBase.camManager
BattleActionsBase.vFXInstancer
BattleActionsBase.battleGameMode
BattleActionsBase.battleUIController
BattleActionsBase.shinseiTypeScriptable
BattleActionsBase.launchVfx
BattleActionsBase.vfxTime
BattleActionsBase.actionTime
BattleActionsBase.targetName
BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)
BattleActionsBase.CheckAction(Single, Action)
BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)
BattleActionsBase.ShowBothShinseisFar()
BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)
BattleActionsBase.ActionTime()

Namespace: Global
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionPutAlteredState : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData putAlteredStateData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	putAlteredStateData	

Overrides

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData putAlteredStateData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	putAlteredStateData	
Action	onEndVfxCallback	

Overrides

BattleActionsBase.ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Class BattleActionReflect

This class controls a special attack case when shinsei reflect damage

Inheritance

System.Object

BattleActionsBase

BattleActionReflect

Inherited Members

BattleActionsBase.actionType

BattleActionsBase.camManager

BattleActionsBase.vFXInstancer

BattleActionsBase.battleGameMode

BattleActionsBase.battleUIController

BattleActionsBase.shinseiTypeScriptable

BattleActionsBase.launchVfx

BattleActionsBase.vfxTime

BattleActionsBase.actionTime

BattleActionsBase.targetName

BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)

BattleActionsBase.CheckAction(Single, Action)

BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)

BattleActionsBase.ShowBothShinseisFar()

BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)

BattleActionsBase.ActionTime()

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Namespace: Global

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionReflect : BattleActionsBase
```

Methods

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData battleActionData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	battleActionData	

TYPE	NAME	DESCRIPTION
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionsBase

Inheritance

- System.Object
- BattleActionsBase
- [BattleActionBlockCard](#)
- [BattleActionBuffDebuff](#)
- [BattleActionChangeShinsei](#)
- [BattleActionChangeTerrain](#)
- [BattleActionDamage](#)
- [BattleActionHeal](#)
- [BattleActionPutAlteredState](#)
- [BattleActionReflect](#)
- [BattleActionsBaseTemplate](#)
- [BattleActionSkipTurn](#)
- [BattleActionStatSwap](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public abstract class BattleActionsBase : MonoBehaviour
```

Fields

actionTime

Declaration

```
protected float actionTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

actionType

Declaration

```
public ActionTypeEnum actionType
```

Field Value

TYPE	DESCRIPTION
ActionTypeEnum	

battleGameMode

Declaration

```
protected BattleGameMode battleGameMode
```

Field Value

TYPE	DESCRIPTION
BattleGameMode	

battleUIController

Declaration

protected BattleUIController battleUIController

Field Value

TYPE	DESCRIPTION
BattleUIController	

camManager

Declaration

protected CameraPlaneController camManager
--

Field Value

TYPE	DESCRIPTION
CameraPlaneController	

launchVfx

Declaration

protected bool launchVfx

Field Value

TYPE	DESCRIPTION
System.Boolean	

shinseiTypeScriptable

Declaration

protected ShinseiTypeScriptable shinseiTypeScriptable

Field Value

TYPE	DESCRIPTION
ShinseiTypeScriptable	

targetName

Declaration

protected string targetName

Field Value

TYPE	DESCRIPTION
System.String	

vFXInstancer

Declaration

protected VFXInstancer vFXInstancer

Field Value

TYPE	DESCRIPTION
VFXInstancer	

vfxTime

Declaration

protected float vfxTime

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

ActionTime()

Declaration

public virtual float ActionTime()

Returns

TYPE	DESCRIPTION
System.Single	

CheckAction(Single, Action)

Declaration

public IEnumerator CheckAction(float time, Action onEndVfxCallback)

Parameters

TYPE	NAME	DESCRIPTION
System.Single	time	
Action	onEndVfxCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

DecideIndexes(out VFXPositionEnum, out VFXPositionEnum, out CamerasAvailableEnum, out CamerasAvailableEnum, Boolean, BattleActionData)

Declaration

```
public void DecideIndexes(out VFXPositionEnum ownerPositionIndexForMuzzle, out VFXPositionEnum targetPositionIndex, out CamerasAvailableEnum lookAtShinseiCam, out CamerasAvailableEnum lookAtOtherShinseiCam, bool isLocal, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
VFXPositionEnum	ownerPositionIndexForMuzzle	
VFXPositionEnum	targetPositionIndex	
CamerasAvailableEnum	lookAtShinseiCam	
CamerasAvailableEnum	lookAtOtherShinseiCam	
System.Boolean	isLocal	
BattleActionData	battleActionData	

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public virtual void EndAction(bool isLocalPlayer, UserInfo ownerPlayerAction, UserInfo otherPlayer, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayerAction	
UserInfo	otherPlayer	
BattleActionData	battleActionData	

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public virtual void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData battleActionData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	battleActionData	
Action	onEndVfxCallback	

Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)

Declaration

```
public virtual void Init(CameraPlaneController _camManager, VFXInstancer _vFXInstancer, BattleGameMode _battleGameMode, BattleUIController _battleUIController, ShinseiTypeScriptable _shinseiTypeScriptable)
```

Parameters

TYPE	NAME	DESCRIPTION
CameraPlaneController	_camManager	
VFXInstancer	_vFXInstancer	
BattleGameMode	_battleGameMode	
BattleUIController	_battleUIController	
ShinseiTypeScriptable	_shinseiTypeScriptable	

PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)

Declaration

```
public void PlayAnimation(UserInfo targetInfo, AttacksAnimation animation, bool isLocalPlayer, BattleActionData battleActionData)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	targetInfo	
AttacksAnimation	animation	
System.Boolean	isLocalPlayer	
BattleActionData	battleActionData	

ShowBothShinseisFar()

Declaration

```
public void ShowBothShinseisFar()
```

Class BattleActionsBaseTemplate

Inheritance

System.Object

BattleActionsBase

BattleActionsBaseTemplate

Inherited Members

BattleActionsBase.actionType

BattleActionsBase.camManager

BattleActionsBase.vFXInstancer

BattleActionsBase.battleGameMode

BattleActionsBase.battleUIController

BattleActionsBase.shinseiTypeScriptable

BattleActionsBase.launchVfx

BattleActionsBase.vfxTime

BattleActionsBase.actionTime

BattleActionsBase.targetName

BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)

BattleActionsBase.ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

BattleActionsBase.CheckAction(Single, Action)

BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)

BattleActionsBase.ShowBothShinseisFar()

BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)

BattleActionsBase.ActionTime()

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionsBaseTemplate : BattleActionsBase
```

Class BattleActionSkipTurn

This class allow user send a turn withouth data

Inheritance

System.Object
BattleActionsBase
BattleActionSkipTurn

Inherited Members

BattleActionsBase.actionType
BattleActionsBase.camManager
BattleActionsBase.vFXInstancer
BattleActionsBase.battleGameMode
BattleActionsBase.battleUIController
BattleActionsBase.shinseiTypeScriptable
BattleActionsBase.launchVfx
BattleActionsBase.vfxTime
BattleActionsBase.actionTime
BattleActionsBase.targetName
BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)
BattleActionsBase.CheckAction(Single, Action)
BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)
BattleActionsBase.ShowBothShinseisFar()
BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)
BattleActionsBase.ActionTime()

Namespace: Global
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionSkipTurn : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData putAlteredStateData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	putAlteredStateData	

Overrides

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData skipTurnData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	skipTurnData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleActionStatSwap

This class allow swap between two stats

Inheritance

System.Object
BattleActionsBase
BattleActionStatSwap

Inherited Members

BattleActionsBase.actionType
BattleActionsBase.camManager
BattleActionsBase.vFXInstancer
BattleActionsBase.battleGameMode
BattleActionsBase.battleUIController
BattleActionsBase.shinseiTypeScriptable
BattleActionsBase.launchVfx
BattleActionsBase.vfxTime
BattleActionsBase.actionTime
BattleActionsBase.targetName
BattleActionsBase.Init(CameraPlaneController, VFXInstancer, BattleGameMode, BattleUIController, ShinseiTypeScriptable)
BattleActionsBase.CheckAction(Single, Action)
BattleActionsBase.DecideIndexes(VFXPositionEnum, VFXPositionEnum, CamerasAvailableEnum, CamerasAvailableEnum, Boolean, BattleActionData)
BattleActionsBase.ShowBothShinseisFar()
BattleActionsBase.PlayAnimation(UserInfo, AttacksAnimation, Boolean, BattleActionData)
BattleActionsBase.ActionTime()

Namespace: Global
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleActionStatSwap : BattleActionsBase
```

Methods

EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

Declaration

```
public override void EndAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData statSwapData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	statSwapData	

Overrides

BattleActionsBase.EndAction(Boolean, UserInfo, UserInfo, BattleActionData)

ExecuteAction(Boolean, UserInfo, UserInfo, BattleActionData, Action)

Declaration

```
public override void ExecuteAction(bool isLocalPlayer, UserInfo ownerPlayer, UserInfo otherPlayer, BattleActionData statSwapData, Action onEndVfxCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	
UserInfo	ownerPlayer	
UserInfo	otherPlayer	
BattleActionData	statSwapData	
Action	onEndVfxCallback	

Overrides

[BattleActionsBase.ExecuteAction\(Boolean, UserInfo, UserInfo, BattleActionData, Action\)](#)

Class BattleAlteredStateBase

Inheritance

System.Object
BattleAlteredStateBase
[BattleAlteredStateBleeding](#)
[BattleAlteredStateEvasionChange](#)
[BattleAlteredStateIgnited](#)
[BattleAlteredStateRoot](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public abstract class BattleAlteredStateBase : MonoBehaviour
```

Fields

alteredStateData

Declaration

```
public BattleAlteredStateDataSO alteredStateData
```

Field Value

TYPE	DESCRIPTION
BattleAlteredStateDataSO	

Methods

EndAlteredState(UserInfo, Boolean)

Declaration

```
public virtual void EndAlteredState(UserInfo _targetInfo = null, bool isTargetLocalPlayer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	_targetInfo	
System.Boolean	isTargetLocalPlayer	

ExecuteAlteredState(UserInfo, List<BattleActionData>, Boolean)

Declaration

```
public virtual void ExecuteAlteredState(UserInfo _targetInfo = null, List<BattleActionData> turnActions = null, bool isTargetLocalPlayer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	_targetInfo	
System.Collections.Generic.List< BattleActionData >	turnActions	

TYPE	NAME	DESCRIPTION
System.Boolean	isTargetLocalPlayer	

InitAlteredState(Boolean, UserInfo, BattleGameMode, BattleUIController)

Declaration

```
public virtual void InitAlteredState(bool isLocal, UserInfo _targetInfo, BattleGameMode _battleGameMode, BattleUIController _battleUIController)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocal	
UserInfo	_targetInfo	
BattleGameMode	_battleGameMode	
BattleUIController	_battleUIController	

Class BattleAlteredStateBleeding

Inheritance

System.Object

[BattleAlteredStateBase](#)

BattleAlteredStateBleeding

Inherited Members

[BattleAlteredStateBase.alteredStateData](#)

[BattleAlteredStateBase.InitAlteredState\(Boolean, UserInfo, BattleGameMode, BattleUIController\)](#)

[BattleAlteredStateBase.ExecuteAlteredState\(UserInfo, List<BattleActionData>, Boolean\)](#)

[BattleAlteredStateBase.EndAlteredState\(UserInfo, Boolean\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleAlteredStateBleeding : BattleAlteredStateBase
```

Class BattleAlteredStateDataSO

Inheritance

System.Object

BattleAlteredStateDataSO

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class BattleAlteredStateDataSO : SerializedScriptableObject
```

Fields

alteredState

Declaration

```
public AlteredStateEnum alteredState
```

Field Value

TYPE	DESCRIPTION
AlteredStateEnum	

alteredStateActions

Declaration

```
public ActionDataList alteredStateActions
```

Field Value

TYPE	DESCRIPTION
ActionDataList	

displayMessage

Declaration

```
public string displayMessage
```

Field Value

TYPE	DESCRIPTION
System.String	

endMessage

Declaration

```
public string endMessage
```

Field Value

TYPE	DESCRIPTION
System.String	

Class BattleAlteredStateEvasionChange

Inheritance

System.Object
[BattleAlteredStateBase](#)
BattleAlteredStateEvasionChange

Inherited Members

[BattleAlteredStateBase.alteredStateData](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleAlteredStateEvasionChange : BattleAlteredStateBase
```

Methods

EndAlteredState(UserInfo, Boolean)

Declaration

```
public override void EndAlteredState(UserInfo _targetInfo = null, bool isTargetLocalPlayer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	_targetInfo	
System.Boolean	isTargetLocalPlayer	

Overrides

[BattleAlteredStateBase.EndAlteredState\(UserInfo, Boolean\)](#)

ExecuteAlteredState(UserInfo, List<BattleActionData>, Boolean)

Declaration

```
public override void ExecuteAlteredState(UserInfo _targetInfo = null, List<BattleActionData> turnActions = null, bool isTargetLocalPlayer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	_targetInfo	
System.Collections.Generic.List< BattleActionData >	turnActions	
System.Boolean	isTargetLocalPlayer	

Overrides

[BattleAlteredStateBase.ExecuteAlteredState\(UserInfo, List<BattleActionData>, Boolean\)](#)

InitAlteredState(Boolean, UserInfo, BattleGameMode, BattleUIController)

Declaration

```
public override void InitAlteredState(bool isLocal, UserInfo _targetInfo, BattleGameMode _battleGameMode, BattleUIController _battleUIController)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocal	
UserInfo	_targetInfo	
BattleGameMode	_battleGameMode	
BattleUIController	_battleUIController	

Overrides

[BattleAlteredStateBase.InitAlteredState\(Boolean, UserInfo, BattleGameMode, BattleUIController\)](#)

Class BattleAlteredStateIgnited

Specific case of Altered State Ignite

Inheritance

System.Object

[BattleAlteredStateBase](#)

BattleAlteredStateIgnited

Inherited Members

[BattleAlteredStateBase.alteredStateData](#)

[BattleAlteredStateBase.InitAlteredState\(Boolean, UserInfo, BattleGameMode, BattleUIController\)](#)

[BattleAlteredStateBase.EndAlteredState\(UserInfo, Boolean\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleAlteredStateIgnited : BattleAlteredStateBase
```

Methods

ExecuteAlteredState(UserInfo, List<BattleActionData>, Boolean)

Declaration

```
public override void ExecuteAlteredState(UserInfo _targetInfo = null, List<BattleActionData> turnActions = null, bool isTargetLocalPlayer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	_targetInfo	
System.Collections.Generic.List< BattleActionData >	turnActions	
System.Boolean	isTargetLocalPlayer	

Overrides

[BattleAlteredStateBase.ExecuteAlteredState\(UserInfo, List<BattleActionData>, Boolean\)](#)

Class BattleAlteredStateRoot

The specific case of altered state Root

Inheritance

System.Object
[BattleAlteredStateBase](#)
BattleAlteredStateRoot

Inherited Members

[BattleAlteredStateBase.alteredStateData](#)
[BattleAlteredStateBase.InitAlteredState\(Boolean, UserInfo, BattleGameMode, BattleUIController\)](#)
[BattleAlteredStateBase.EndAlteredState\(UserInfo, Boolean\)](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleAlteredStateRoot : BattleAlteredStateBase
```

Methods

ExecuteAlteredState(UserInfo, List<BattleActionData>, Boolean)

Declaration

```
public override void ExecuteAlteredState(UserInfo ownerInfo = null, List<BattleActionData> turnActions = null, bool isTargetLocalPlayer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	ownerInfo	
System.Collections.Generic.List< BattleActionData >	turnActions	
System.Boolean	isTargetLocalPlayer	

Overrides

[BattleAlteredStateBase.ExecuteAlteredState\(UserInfo, List<BattleActionData>, Boolean\)](#)

Class BattleNotificationSystem

Inheritance

System.Object

BattleNotificationSystem

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleNotificationSystem : TextboxBase
```

Class BattleTerrainBehavioursBase

Inheritance

System.Object
BattleTerrainBehavioursBase
[BattleTerrainBehaviourSnow](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public abstract class BattleTerrainBehavioursBase : MonoBehaviour
```

Fields

allShinseis

Declaration

```
protected List<Shinsei> allShinseis
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

battleGameMode

Declaration

```
protected BattleGameMode battleGameMode
```

Field Value

TYPE	DESCRIPTION
BattleGameMode	

terrainData

Declaration

```
public BattleTerrainDataSO terrainData
```

Field Value

TYPE	DESCRIPTION
BattleTerrainDataSO	

turnsDuration

Declaration

```
protected int turnsDuration
```

Field Value

TYPE	DESCRIPTION
System.Int32	

turnsLeft

Declaration

public int turnsLeft

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

EndAction(CharacterType, Boolean)

Declaration

public float EndAction(CharacterType shinseiType, bool isPlayer = false)
--

Parameters

TYPE	NAME	DESCRIPTION
CharacterType	shinseiType	
System.Boolean	isPlayer	

Returns

TYPE	DESCRIPTION
System.Single	

EndTerrainBehaviour()

End terrain behaviour, return time of behaviour

Declaration

public virtual float EndTerrainBehaviour()
--

Returns

TYPE	DESCRIPTION
System.Single	

ExecuteAction(CharacterType, Boolean)

Declaration

public float ExecuteAction(CharacterType shinseiType, bool isPlayer = false)
--

Parameters

TYPE	NAME	DESCRIPTION
CharacterType	shinseiType	
System.Boolean	isPlayer	

Returns

TYPE	DESCRIPTION
System.Single	

ExecuteActionsOfTerrain(Func<CharacterType, Boolean, Single>, Boolean)

Execute the actions set to this terrain,

Declaration

```
public virtual float ExecuteActionsOfTerrain(Func<CharacterType, bool, float> executeOrEnd, bool toAll = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Func<CharacterType, System.Boolean, System.Single>	executeOrEnd	
System.Boolean	toAll	toAll shinseis or just to the current ones

Returns

TYPE	DESCRIPTION
System.Single	

ExecuteTerrainBehaviour()

Execute terrain behaviour, return time of behaviour

Declaration

```
public virtual float ExecuteTerrainBehaviour()
```

Returns

TYPE	DESCRIPTION
System.Single	

InitTerrainBehaviour(BattleGameMode)

Init terrain behaviour, return time of behaviour

Declaration

```
public virtual float InitTerrainBehaviour(BattleGameMode battleGameMode)
```

Parameters

TYPE	NAME	DESCRIPTION
BattleGameMode	battleGameMode	

Returns

TYPE	DESCRIPTION
System.Single	

Class BattleTerrainBehaviourSnow

Inheritance

System.Object
[BattleTerrainBehavioursBase](#)
BattleTerrainBehaviourSnow

Inherited Members

- [BattleTerrainBehavioursBase.allShinseis](#)
- [BattleTerrainBehavioursBase.turnsLeft](#)
- [BattleTerrainBehavioursBase.turnsDuration](#)
- [BattleTerrainBehavioursBase.battleGameMode](#)
- [BattleTerrainBehavioursBase.terrainData](#)
- [BattleTerrainBehavioursBase.ExecuteActionsOfTerrain\(Func<CharacterType, Boolean, Single>, Boolean\)](#)
- [BattleTerrainBehavioursBase.ExecuteAction\(CharacterType, Boolean\)](#)
- [BattleTerrainBehavioursBase.EndAction\(CharacterType, Boolean\)](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class BattleTerrainBehaviourSnow : BattleTerrainBehavioursBase
```

Methods

EndTerrainBehaviour()

Declaration

```
public override float EndTerrainBehaviour()
```

Returns

TYPE	DESCRIPTION
System.Single	

Overrides

[BattleTerrainBehavioursBase.EndTerrainBehaviour\(\)](#)

ExecuteTerrainBehaviour()

Declaration

```
public override float ExecuteTerrainBehaviour()
```

Returns

TYPE	DESCRIPTION
System.Single	

Overrides

[BattleTerrainBehavioursBase.ExecuteTerrainBehaviour\(\)](#)

InitTerrainBehaviour(BattleGameMode)

Declaration

```
public override float InitTerrainBehaviour(BattleGameMode battleGameMode)
```

Parameters

TYPE	NAME	DESCRIPTION
BattleGameMode	battleGameMode	

Returns

TYPE	DESCRIPTION
System.Single	

Overrides

[BattleTerrainBehavioursBase.InitTerrainBehaviour\(BattleGameMode\)](#)

Class BattleTerrainDataSO

Inheritance

System.Object

BattleTerrainDataSO

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class BattleTerrainDataSO : SerializedScriptableObject
```

Fields

displayMessage

Declaration

```
public string displayMessage
```

Field Value

TYPE	DESCRIPTION
System.String	

globalActions

Declaration

```
public ActionDataList globalActions
```

Field Value

TYPE	DESCRIPTION
ActionDataList	

terrainType

Declaration

```
public TypesOfTerrainEnum terrainType
```

Field Value

TYPE	DESCRIPTION
TypesOfTerrainEnum	

typesActions

Declaration

```
public List<TypesActions> typesActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TypesActions >	

Methods

GetJsonActionCards()

Declaration

```
public void GetJsonActionCards()
```

Class BodyStyle

Inheritance

System.Object
BodyStyle

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class BodyStyle : MonoBehaviour
```

Fields

bodyParts

Declaration

```
public List<BodyStyle.BodyPartDressable> bodyParts
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BodyStyle.BodyPartDressable >	

Class BodyStyle.BodyPartDressable

Inheritance

System.Object
BodyStyle.BodyPartDressable

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class BodyPartDressable
```

Methods

SelectObject(Int32, Boolean)

Declaration

```
public void SelectObject(int index, bool isLocal = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	
System.Boolean	isLocal	

Class BoneAnimationTracker

Inheritance

System.Object

BoneAnimationTracker

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BoneAnimationTracker : MonoBehaviour
```

Fields

boneTracker

Declaration

```
public GameObject boneTracker
```

Field Value

TYPE	DESCRIPTION
GameObject	

matAnimations

Declaration

```
public List<MatAnimation> matAnimations
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< MatAnimation >	

materials

Declaration

```
public List<Material> materials
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Material>	

Methods

Update()

Declaration

```
public void Update()
```

VerifyAllMatAnimations(MatAnimation.AttachedEdge, Single)

Declaration

```
public void VerifyAllMatAnimations(MatAnimation.AttachedEdge targetAttachedEdge, float bonePositionEdge)
```

Parameters

TYPE	NAME	DESCRIPTION
MatAnimation.AttachedEdge	targetAttachedEdge	
System.Single	bonePositionEdge	

Class BracketsData

Inheritance

System.Object

BracketsData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BracketsData
```

Fields

displayName1

Declaration

```
public string displayName1
```

Field Value

TYPE	DESCRIPTION
System.String	

displayName2

Declaration

```
public string displayName2
```

Field Value

TYPE	DESCRIPTION
System.String	

matchId

Declaration

```
public string matchId
```

Field Value

TYPE	DESCRIPTION
System.String	

p1Joined

Declaration

```
public bool p1Joined
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

p2Joined

Declaration

```
public bool p2Joined
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

PID1

Declaration

```
public string PID1
```

Field Value

TYPE	DESCRIPTION
System.String	

PID2

Declaration

```
public string PID2
```

Field Value

TYPE	DESCRIPTION
System.String	

Class BracketsTournamentManager

Inheritance

System.Object

BracketsTournamentManager

Implements

[IBracketsTournament](#)

IService

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BracketsTournamentManager : MonoBehaviour, IBracketsTournament
```

Fields

alreadyRecheckConnection

Declaration

```
public bool alreadyRecheckConnection
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

checkTournamentInscription

Declaration

```
public CheckTournamentInscription checkTournamentInscription
```

Field Value

TYPE	DESCRIPTION
CheckTournamentInscription	

checkTournamentStateController

Declaration

```
public CheckTournamentStateController checkTournamentStateController
```

Field Value

TYPE	DESCRIPTION
CheckTournamentStateController	

currentStage

Declaration

```
public int currentStage
```

Field Value

TYPE	DESCRIPTION
System.Int32	

tournamentBracketsShowController

Declaration

```
public TournamentBracketsShowController tournamentBracketsShowController
```

Field Value

TYPE	DESCRIPTION
TournamentBracketsShowController	

Properties

CheckTournamentInscription

Declaration

```
public CheckTournamentInscription CheckTournamentInscription { get; }
```

Property Value

TYPE	DESCRIPTION
CheckTournamentInscription	

CheckTournamentStateController

Declaration

```
public CheckTournamentStateController CheckTournamentStateController { get; }
```

Property Value

TYPE	DESCRIPTION
CheckTournamentStateController	

Methods

Awake()

Declaration

```
public void Awake()
```

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

RecheckConection()

Declaration

```
public void RecheckConection()
```

SetAlreadyConnection(Boolean)

Declaration

```
public void SetAlreadyConnection(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

ShowPanelBracketsView(Boolean)

Declaration

```
public void ShowPanelBracketsView(bool active)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	active	

Implements

[IBracketsTournament](#)

IService

Enum CameraPhasesEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum CameraPhasesEnum
```

Fields

NAME	DESCRIPTION
PHASE_A	
PHASE_B	
PHASE_C	
PHASE_D	

Enum CameraPointOfInteresEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum CameraPointOfInteresEnum
```

Fields

NAME	DESCRIPTION
ARENA_CENTER	
ENEMY_SHINSEI	
PLAYER_SHINSEI	

Enum CamerasAvailableEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum CamerasAvailableEnum
```

Fields

NAME	DESCRIPTION
CENITAL_ENEMY	
CENITAL_PLAYER	
DEFAULT_CAMERA	
FAR_MIDDLE_CAMERA	
GENERAL_CAMERA	
LOOK_AT_CAMERA_ENEMY	
LOOK_AT_CAMERA_PLAYER	
MIDDLE_CAMERA	
ORBIT_CAMERA	
ORBIT_CAMERA_BACK	
PAN_CAMERA	
PAN_CAMERA_BACK	
SIDE_CAMERA_ENEMY	
SIDE_CAMERA_PLAYER	

Class CameraSpawner

Inheritance

System.Object

CameraSpawner

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CameraSpawner : MonoBehaviour
```

Class ChallengePlayerController

Inheritance

System.Object

ChallengePlayerController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChallengePlayerController : MonoBehaviour
```

Fields

isRecivingChallenge

Declaration

```
public bool isRecivingChallenge
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

BackButtonLogic(Boolean)

Declaration

```
public void BackButtonLogic(bool waitCancel = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	waitCancel	

CheckChallenge()

Declaration

```
public void CheckChallenge()
```

ConfirmMatch(String, String, String)

Declaration

```
public void ConfirmMatch(string message, string otherPlayerPlayfabId, string otherPlayerDisplayName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	
System.String	otherPlayerPlayfabId	
System.String	otherPlayerDisplayName	

MatchCanceledByChallenged(Boolean)

Declaration

```
public void MatchCanceledByChallenged(bool showOtherPlayerDeclinePopup)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	showOtherPlayerDeclinePopup	

MatchCanceledWait(Boolean)

Declaration

```
public IEnumerable MatchCanceledWait(bool waitPetitionTime = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	waitPetitionTime	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

RecieveChallenge(ThirdPersonController, String)

Declaration

```
public void RecieveChallenge(ThirdPersonController _otherPlayer, string _otherRandomMatchNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
ThirdPersonController	_otherPlayer	
System.String	_otherRandomMatchNumber	

ResetAlreadyHasChallenge()

Declaration

```
public void ResetAlreadyHasChallenge()
```

TimingAcceptChallenge(String, Dictionary<PopupManager.ButtonType, Action>, Boolean)

Declaration

```
public IEnumerable TimingAcceptChallenge(string mainMessage, Dictionary<PopupManager.ButtonType, Action> mainButtons, bool isRecievingChallenge)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mainMessage	
System.Collections.Generic.Dictionary<PopupManager.ButtonType, Action>	mainButtons	
System.Boolean	isRecievingChallenge	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

WaitforOpponentToConfirm(GetMatchResult)

Declaration

```
public IEnumerator WaitforOpponentToConfirm(GetMatchResult matchData)
```

Parameters

TYPE	NAME	DESCRIPTION
GetMatchResult	matchData	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class ChangeColor

Inheritance

System.Object

ChangeColor

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChangeColor : MonoBehaviour
```

Fields

color

Declaration

```
public Gradient color
```

Field Value

TYPE	DESCRIPTION
Gradient	

m_changeColor

Declaration

```
public Color m_changeColor
```

Field Value

TYPE	DESCRIPTION
Color	

m_ColorHandler

Declaration

```
public Image m_ColorHandler
```

Field Value

TYPE	DESCRIPTION
Image	

m_obj

Declaration

```
public GameObject m_obj
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

ChangeEffectColor(Single)

Declaration

```
public void ChangeEffectColor(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	

CheckColorState()

Declaration

```
public void CheckColorState()
```

CheckIsColorChange(Boolean)

Declaration

```
public void CheckIsColorChange(bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	value	

Class ChangeGender

Change between male and female character and update in playfab data

Inheritance

System.Object

ChangeGender

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChangeGender : MonoBehaviour
```

Methods

SelectGender(Int32)

Declaration

```
public void SelectGender(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Class ChangelconLeague

Inheritance

System.Object

ChangelconLeague

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChangeIconLeague : MonoBehaviour
```

Methods

ChangelconUsingIndex(Int32)

Declaration

```
public void ChangeIconUsingIndex(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Class CharacterAnimBehaviour

Inheritance

System.Object

CharacterAnimBehaviour

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterAnimBehaviour : StateMachineBehaviour
```

Methods

OnStateMachineEnter(Animator, Int32)

Declaration

```
public override void OnStateMachineEnter(Animator animator, int stateMachinePathHash)
```

Parameters

TYPE	NAME	DESCRIPTION
Animator	animator	
System.Int32	stateMachinePathHash	

Class CharacterAPI

Inheritance

System.Object
CharacterAPI

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class CharacterAPI
```

Fields

Accessory

Declaration

```
public long Accessory
```

Field Value

TYPE	DESCRIPTION
System.Int64	

Body

Declaration

```
public long Body
```

Field Value

TYPE	DESCRIPTION
System.Int64	

Ears

Declaration

```
public long Ears
```

Field Value

TYPE	DESCRIPTION
System.Int64	

Head

Declaration

```
public long Head
```

Field Value

TYPE	DESCRIPTION
System.Int64	

Tail

Declaration

public long Tail

Field Value

TYPE	DESCRIPTION
System.Int64	

Class CharacterColorSlot

UI Element that represents a color for character

Inheritance

System.Object

[CharacterStyleSlot](#)

CharacterColorSlot

Inherited Members

[CharacterStyleSlot.OnColorSelected](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterColorSlot : CharacterStyleSlot
```

Fields

colorSlotImage

Declaration

```
public Image colorSlotImage
```

Field Value

TYPE	DESCRIPTION
Image	

Methods

InitSlot<T>(T)

Declaration

```
public override void InitSlot<T>(T _colorValue)
    where T : CharacterStyleRelation
```

Parameters

TYPE	NAME	DESCRIPTION
T	_colorValue	

Type Parameters

NAME	DESCRIPTION
T	

Overrides

CharacterStyleSlot.InitSlot<T>(T)

SelectLockedSlot()

Declaration

```
public void SelectLockedSlot()
```

SelectSlot()

Declaration

```
public void SelectSlot()
```

Class CharacterDatabase

Inheritance

System.Object

CharacterDatabase

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterDatabase : ScriptableObject
```

Properties

CaptureImagePath

Declaration

```
public string CaptureImagePath { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

CharacterSlotPrefabsList

Declaration

```
public List<CharacterSlot> CharacterSlotPrefabsList { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<CharacterSlot>	

NFTsModels

Declaration

```
public NFTsModel[] NFTsModels { get; }
```

Property Value

TYPE	DESCRIPTION
NFTsModel []	

PartEntityModel

Declaration

```
public List<PartEntityModel> PartEntityModel { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< PartEntityModel >	

Methods

AddCharacterSlotToArray(CharacterSlot)

Declaration

```
public void AddCharacterSlotToArray(CharacterSlot _characterSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
CharacterSlot	_characterSlot	

AddElement()

Declaration

```
public void AddElement()
```

AddPartEntityToArray(PartType, PartEntity)

Declaration

```
public void AddPartEntityToArray(PartType _partType, PartEntity _partEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
PartType	_partType	
PartEntity	_partEntity	

DetermineEntityRarity(ref PartEntity)

Declaration

```
public RarityType DetermineEntityRarity(ref PartEntity _partEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
PartEntity	_partEntity	

Returns

TYPE	DESCRIPTION
RarityType	

FindPartEntities(GameObject, Action<GameObject>)

Declaration

```
public void FindPartEntities(GameObject _referenceGameObject, Action<GameObject> _onGetEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_referenceGameObject	
Action<GameObject>	_onGetEntity	

FindPartSlots(GameObject, Action<GameObject>)

Declaration

```
public void FindPartSlots(GameObject _referenceGameObject, Action<GameObject> _onGetSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_referenceGameObject	
Action<GameObject>	_onGetSlot	

FindPartTypeInName(GameObject)

Declaration

```
public PartType FindPartTypeInName(GameObject _referenceGameObject)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_referenceGameObject	

Returns

TYPE	DESCRIPTION
PartType	

GetNFT(String[])

Declaration

```
public NFTsModel[] GetNFT(string[] _nftsIDs)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	_nftsIDs	

Returns

TYPE	DESCRIPTION
NFTsModel[]	

RemoveEmptyCharacterPrefabs()

Declaration

```
public void RemoveEmptyCharacterPrefabs()
```

Class CharacterSelectorController

Inheritance

System.Object

CharacterSelectorController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterSelectorController : MonoBehaviour
```

Fields

characterContainer

Declaration

```
public Transform characterContainer
```

Field Value

TYPE	DESCRIPTION
Transform	

characterSelectorContainer

Declaration

```
public Transform characterSelectorContainer
```

Field Value

TYPE	DESCRIPTION
Transform	

characterSelectorPrefab

Declaration

```
public CharacterSelector characterSelectorPrefab
```

Field Value

TYPE	DESCRIPTION
CharacterSelector	

partSelectorContainer

Declaration

```
public Transform partSelectorContainer
```

Field Value

TYPE	DESCRIPTION
Transform	

partSelectorPrefab

Declaration

```
public PartSelector partSelectorPrefab
```

Field Value

TYPE	DESCRIPTION
PartSelector	

randomButton

Declaration

```
public Button randomButton
```

Field Value

TYPE	DESCRIPTION
Button	

Methods

GetCharacterUniqueID()

Declaration

```
public void GetCharacterUniqueID()
```

OnButtonPress(Boolean, PartSlot, PartSelector)

Declaration

```
public void OnButtonPress(bool _isLeftButton, PartSlot _dummyBodyPart, PartSelector _dummyPartSelector)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	_isLeftButton	
PartSlot	_dummyBodyPart	
PartSelector	_dummyPartSelector	

OnRandomSelect()

Declaration

```
public void OnRandomSelect()
```

SelectCharacter(Int32)

Declaration

```
public void SelectCharacter(int _characterIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	_characterIndex	

Class CharacterStyleInfo

Inheritance

System.Object
CharacterStyleInfo

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterStyleInfo
```

Fields

colorHex

Declaration

```
public string colorHex
```

Field Value

TYPE	DESCRIPTION
System.String	

presetId

Declaration

```
public int presetId
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class CharacterStyleRelation

Inheritance

System.Object
CharacterStyleRelation
[ColorIdRelation](#)
[PartIdRelation](#)

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterStyleRelation
```

Fields

id

Declaration

```
public int id
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class CharacterStyleSlot

Inheritance

System.Object

CharacterStyleSlot

[CharacterColorSlot](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class CharacterStyleSlot : MonoBehaviour
```

Fields

OnColorSelected

Declaration

```
public Action<CharacterStyleRelation> OnColorSelected
```

Field Value

TYPE	DESCRIPTION
Action< CharacterStyleRelation >	

Methods

InitSlot<T>(T)

Declaration

```
public abstract void InitSlot<T>(T _colorValue)
    where T : CharacterStyleRelation
```

Parameters

TYPE	NAME	DESCRIPTION
T	_colorValue	

Type Parameters

NAME	DESCRIPTION
T	

Struct ChatBadWordsDB

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public struct ChatBadWordsDB
```

Fields

listOfBadWords

Declaration

```
public List<List<string>> listOfBadWords
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.List<System.String>>	

Class ChatTextBox

Chat behavior, send messages and show in a UI Elements

Inheritance

System.Object

ChatTextBox

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChatTextBox : TextboxBase
```

Fields

alreadyWriteMessages

Declaration

```
public List<string> alreadyWriteMessages
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

badWordsDB

Declaration

```
public ChatBadWordsDB badWordsDB
```

Field Value

TYPE	DESCRIPTION
ChatBadWordsDB	

filter

Declaration

```
public GameObject filter
```

Field Value

TYPE	DESCRIPTION
GameObject	

isReady

Declaration

```
public bool isReady
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

OnEndEditing

Declaration

public Action OnEndEditing

Field Value

TYPE	DESCRIPTION
Action	

OnStartEditing

Declaration

public Action OnStartEditing

Field Value

TYPE	DESCRIPTION
Action	

system

Declaration

public EventSystem system

Field Value

TYPE	DESCRIPTION
EventSystem	

Methods

CallEndEditing()

Declaration

public void CallEndEditing()

CallStartEditing()

Declaration

public void CallStartEditing()

CheckForBadWords(String)

Declaration

public string CheckForBadWords(string message)
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

Returns

TYPE	DESCRIPTION
System.String	

DisableMovement(Boolean)

Declaration

```
public void DisableMovement(bool newState)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	newState	

SendLocalMessage()

Declaration

```
public void SendLocalMessage()
```

SendMessage(ChatMessagePayload, String)

Declaration

```
public void SendMessage(ChatMessagePayload chatMessage, string from)
```

Parameters

TYPE	NAME	DESCRIPTION
ChatMessagePayload	chatMessage	
System.String	from	

SendMessage(ChatMessagePayload, String, Boolean)

Declaration

```
public void SendMessage(ChatMessagePayload chatMessage, string displayName, bool isAdmin = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ChatMessagePayload	chatMessage	
System.String	displayName	
System.Boolean	isAdmin	

ShowHide(Boolean)

Declaration

```
public void ShowHide(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

ToggleShowFilter()

Declaration

```
public void ToggleShowFilter()
```

Update()

Declaration

```
public void Update()
```

Class CheckStateOfTournamentDto

Inheritance

System.Object

CheckStateOfTournamentDto

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CheckStateOfTournamentDto
```

Fields

alredyPlayedPlayers

Declaration

```
public List<string> alredyPlayedPlayers
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

currentStage

Declaration

```
public int currentStage
```

Field Value

TYPE	DESCRIPTION
System.Int32	

nextRoundInitTime

Declaration

```
public string nextRoundInitTime
```

Field Value

TYPE	DESCRIPTION
System.String	

Class CheckTournamentInscription

Allow hide all objects innecessary in tournament mode, only visual results

Inheritance

System.Object

CheckTournamentInscription

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CheckTournamentInscription : MonoBehaviour
```

Fields

hasAlreadyRecheckConnection

Declaration

```
public bool hasAlreadyRecheckConnection
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

onCheckIfTournamentExist

Declaration

```
public Action onCheckIfTournamentExist
```

Field Value

TYPE	DESCRIPTION
Action	

Methods

DisableObjectsInTournament(Boolean)

Declaration

```
public void DisableObjectsInTournament(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

EnableNPCs()

Declaration

```
public void EnableNPCs()
```

Class CheckTournamentStateController

Inheritance

System.Object

CheckTournamentStateController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CheckTournamentStateController : MonoBehaviour
```

Methods

CheckTournamentState(Nullable<Boolean>, Boolean)

Declaration

```
public void CheckTournamentState(bool? isLocalPlayerWon = null, bool shouldCheckWinByDefault = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<System.Boolean>	isLocalPlayerWon	
System.Boolean	shouldCheckWinByDefault	

WaitForSeconds(Single, Action)

Declaration

```
public IEnumerator WaitForSeconds(float seconds, Action callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	seconds	
Action	callback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class ColorIdRelation

Inheritance

System.Object

[CharacterStyleRelation](#)

ColorIdRelation

Inherited Members

[CharacterStyleRelation.id](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class ColorIdRelation : CharacterStyleRelation
```

Fields

color

Declaration

```
public Color color
```

Field Value

TYPE	DESCRIPTION
Color	

usableOnParts

Declaration

```
public List<PartsOfCharacter> usableOnParts
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PartsOfCharacter>	

Class ColorSO

Inheritance

System.Object

ColorSO

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ColorSO : ScriptableObject
```

Fields

_paletteColor

Declaration

```
public Color32[] _paletteColor
```

Field Value

TYPE	DESCRIPTION
Color32[]	

Class ColorsSO

Inheritance

System.Object

ColorsSO

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ColorsSO : ScriptableObject
```

Fields

_palettes

Declaration

```
public ColorSO[] _palettes
```

Field Value

TYPE	DESCRIPTION
ColorSO []	

Class ColorSwapper

Inheritance

System.Object
Utils.Singleton
Utils.Singleton<[ColorSwapper](#)>
ColorSwapper

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class ColorSwapper : Singleton<ColorSwapper>
```

Fields

[_colorsScriptableObject](#)

Declaration

```
public ColorsSO _colorsScriptableObject
```

Field Value

TYPE	DESCRIPTION
ColorsSO	

[_dontTakePhoto](#)

Declaration

```
public bool _dontTakePhoto
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

[_partMaterials](#)

Declaration

```
public List<Material> _partMaterials
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Material>	

[PaletteID](#)

Declaration

```
public string PaletteID
```

Field Value

TYPE	DESCRIPTION
System.String	

partVisuals

Declaration

```
public List<PartVisual> partVisuals
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< PartVisual >	

Methods

AddItemToPVL(PartVisual)

Declaration

```
public void AddItemToPVL(PartVisual PV)
```

Parameters

TYPE	NAME	DESCRIPTION
PartVisual	PV	

ChangeMaterial()

Declaration

```
public void ChangeMaterial()
```

Class CombatFirebasePetitions

Deprecate battle using firebase

Inheritance

System.Object

CombatFirebasePetitions

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class CombatFirebasePetitions
```

Fields

currentMatchId

Declaration

```
public static string currentMatchId
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

DeleteDataPetition(Action<Boolean>, String)

Declaration

```
public static void DeleteDataPetition(Action<bool> onSendSuccess = null, string key = "")
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onSendSuccess	
System.String	key	

GetDataPetition(Action<Boolean>, Action<Dictionary<String, OrderMessage>>)

Declaration

```
public static void GetDataPetition(Action<bool> onSendSuccess = null, Action<Dictionary<string, OrderMessage>> dataCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onSendSuccess	
Action<System.Collections.Generic.Dictionary<System.String, OrderMessage >>	dataCallback	

SetMatchId(String)

Declaration

```
public static void SetMatchId(string matchId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	

UpdateDataPetition(OrderMessage, Action<Boolean>)

Declaration

```
public static void UpdateDataPetition(OrderMessage message, Action<bool> onSendSuccess = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderMessage	message	
Action<System.Boolean>	onSendSuccess	

Class Constants

Inheritance

System.Object

Constants

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class Constants
```

Fields

CARD_STORE

Declaration

```
public const string CARD_STORE = "CardStore"
```

Field Value

TYPE	DESCRIPTION
System.String	

CHARACTER_STYLE

Declaration

```
public const string CHARACTER_STYLE = "CharacterStyle"
```

Field Value

TYPE	DESCRIPTION
System.String	

FRIENDLY_LEADERBOARD

Declaration

```
public const string FRIENDLY_LEADERBOARD = "Victories"
```

Field Value

TYPE	DESCRIPTION
System.String	

FRIENDLY_MATCH

Declaration

```
public const string FRIENDLY_MATCH = "FriendlyMatch"
```

Field Value

TYPE	DESCRIPTION
System.String	

SHINSEI_COMPANION

Declaration

```
public const string SHINSEI_COMPANION = "ShinseiCompanion"
```

Field Value

TYPE	DESCRIPTION
System.String	

SHINSEI_SLOT

Declaration

```
public const string SHINSEI_SLOT = "ShinseiSlot"
```

Field Value

TYPE	DESCRIPTION
System.String	

SHINSEI_VAULT

Declaration

```
public const string SHINSEI_VAULT = "ShinseiVault"
```

Field Value

TYPE	DESCRIPTION
System.String	

TITLE_ID

Declaration

```
public const string TITLE_ID = "BFDDE"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class Cooker

This script controls the behavior of npc cooker in the bar

Inheritance

System.Object

Cooker

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Cooker : MonoBehaviour
```

Fields

layerMaskForCook

Declaration

```
public LayerMask layerMaskForCook
```

Field Value

TYPE	DESCRIPTION
LayerMask	

Methods

GetFood(Int32)

Playback the cooking animations

Declaration

```
public void GetFood(int type)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	type	

Hide()

Put camera settings to normal values

Declaration

```
public void Hide()
```

Show()

Set camera setting for cooking event

Declaration

```
public void Show()
```

Class Curtain

A component that make a black transition

Inheritance

System.Object

Curtain

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Curtain : MonoBehaviour
```

Methods

StartFade(Int32)

Declaration

```
public void StartFade(int transitionDuration)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	transitionDuration	

Enum CriticsCheck

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum CriticsCheck
```

Fields

NAME	DESCRIPTION
Is_Sleep	

Class csEffect_03_ArrowFunction

Inheritance

System.Object
csEffect_03_ArrowFunction

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_03_ArrowFunction : MonoBehaviour
```

Fields

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lerpValue

Declaration

```
public float m_lerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeObj

Declaration

```
public GameObject m_makeObj
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_objectDestroyTime

Declaration

```
public float m_objectDestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_origianlPosSet

Declaration

```
public Vector3 m_origianlPosSet
```

Field Value

TYPE	DESCRIPTION
Vector3	

Class csEffect_03_ObjectMake

Inheritance

System.Object
csEffect_03_ObjectMake

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_03_ObjectMake : MonoBehaviour
```

Fields

m_destroyTime

Declaration

```
public float m_destroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeCount

Declaration

```
public int m_makeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_makeDelay

Declaration

```
public float m_makeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeObjs

Declaration

```
public GameObject[] m_makeObjs
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

m_makePos

Declaration

```
public Transform m_makePos
```

Field Value

TYPE	DESCRIPTION
Transform	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_vectorRandom

Declaration

```
public Vector3 m_vectorRandom
```

Field Value

TYPE	DESCRIPTION
Vector3	

Class csEffect_76_Animation

Inheritance

System.Object
csEffect_76_Animation

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_76_Animation : MonoBehaviour
```

Fields

m_downYPos

Declaration

```
public float m_downYPos
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_firstLerpValue

Declaration

```
public float m_firstLerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lookAtTime

Declaration

```
public float m_lookAtTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_onlyYValueMove

Declaration

```
public bool m_onlyYValueMove
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_particleDestroyTime

Declaration

```
public float m_particleDestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_particleObject

Declaration

```
public GameObject m_particleObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_secondLerpValue

Declaration

```
public float m_secondLerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_secondMovePos

Declaration

```
public Vector3 m_secondMovePos
```

Field Value

TYPE	DESCRIPTION
Vector3	

m_secondMoveTime

Declaration

```
public float m_secondMoveTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_y_randomValue

Declaration

```
public float m_y_randomValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_y_StopPos

Declaration

```
public float m_y_StopPos
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csEffect_76_CircleSwordMake

Inheritance

System.Object

csEffect_76_CircleSwordMake

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_76_CircleSwordMake : MonoBehaviour
```

Fields

m_DestroyTime

Declaration

```
public float m_DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_interval

Declaration

```
public float m_interval
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makePoint

Declaration

```
public Transform m_makePoint
```

Field Value

TYPE	DESCRIPTION
Transform	

m_object_MakeCount

Declaration

```
public int m_object_MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_object_MakeDelay

Declaration

```
public float m_object_MakeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_yAddedPos

Declaration

```
public float m_yAddedPos
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csEffect_76_FloatingSwordMake

Inheritance

System.Object

csEffect_76_FloatingSwordMake

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_76_FloatingSwordMake : MonoBehaviour
```

Fields

m_DestroyTime

Declaration

```
public float m_DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_makePoint

Declaration

```
public Transform m_makePoint
```

Field Value

TYPE	DESCRIPTION
Transform	

m_movePoint

Declaration

```
public Transform m_movePoint
```

Field Value

TYPE	DESCRIPTION
Transform	

m_object_MakeCount

Declaration

```
public int m_object_MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_object_MakeDelay

Declaration

```
public float m_object_MakeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_randomValue

Declaration

```
public Vector3 m_randomValue
```

Field Value

TYPE	DESCRIPTION
Vector3	

Class csEffect_77_HandMove

Inheritance

System.Object
csEffect_77_HandMove

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_77_HandMove : MonoBehaviour
```

Fields

m_lerpValue

Declaration

```
public float m_lerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_movePos

Declaration

```
public Transform m_movePos
```

Field Value

TYPE	DESCRIPTION
Transform	

Class csEffect_77_ObjectMake

Inheritance

System.Object
csEffect_77_ObjectMake

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_77_ObjectMake : MonoBehaviour
```

Fields

m_addedPos

Declaration

```
public Vector3 m_addedPos
```

Field Value

TYPE	DESCRIPTION
Vector3	

m_destroyTime

Declaration

```
public float m_destroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeDelay

Declaration

```
public float m_makeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeDuration

Declaration

```
public float m_makeDuration
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeObj

Declaration

```
public GameObject m_makeObj
```

Field Value

TYPE	DESCRIPTION
GameObject	

Class csEffect_77_ObjectMake2

Inheritance

System.Object
csEffect_77_ObjectMake2

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_77_ObjectMake2 : MonoBehaviour
```

Fields

m_destroyTime

Declaration

```
public float m_destroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeObj

Declaration

```
public GameObject m_makeObj
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csEffect_79_CircleMake

Inheritance

System.Object

csEffect_79_CircleMake

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_79_CircleMake : MonoBehaviour
```

Fields

m_DestroyTime

Declaration

```
public float m_DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_interval

Declaration

```
public float m_interval
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makePoint

Declaration

```
public Transform m_makePoint
```

Field Value

TYPE	DESCRIPTION
Transform	

m_object_MakeCount

Declaration

```
public int m_object_MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_object_MakeDelay

Declaration

```
public float m_object_MakeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_yAddedPos

Declaration

```
public float m_yAddedPos
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csEffect_79_WallRiseDown

Inheritance

System.Object

csEffect_79_WallRiseDown

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_79_WallRiseDown : MonoBehaviour
```

Fields

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lerpTime

Declaration

```
public float m_lerpTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_y_Pos

Declaration

```
public float m_y_Pos
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csEffect_80_SunMove

Inheritance

System.Object

csEffect_80_SunMove

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csEffect_80_SunMove : MonoBehaviour
```

Fields

DestroyTime

Declaration

```
public float DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lerpValue

Declaration

```
public float m_lerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_movePos

Declaration

```
public Transform m_movePos
```

Field Value

TYPE	DESCRIPTION
Transform	

Class csEffectScene

Inheritance

System.Object
csEffectScene

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class csEffectScene : MonoBehaviour
```

Fields

inputLocation

Declaration

```
public static int inputLocation
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_destroyObjects

Declaration

```
public static GameObject[] m_destroyObjects
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

m_effectName

Declaration

```
public Text m_effectName
```

Field Value

TYPE	DESCRIPTION
Text	

m_effects

Declaration

```
public Transform[] m_effects
```

Field Value

TYPE	DESCRIPTION
Transform[]	

m_explain

Declaration

```
public Text m_explain
```

Field Value

TYPE	DESCRIPTION
Text	

m_packageName

Declaration

```
public Text m_packageName
```

Field Value

TYPE	DESCRIPTION
Text	

m_version

Declaration

```
public Text m_version
```

Field Value

TYPE	DESCRIPTION
Text	

Class csLaser

Inheritance

System.Object

csLaser

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csLaser : MonoBehaviour
```

Fields

Material

Declaration

```
public Material _Material
```

Field Value

TYPE	DESCRIPTION
Material	

LaserHitEffect

Declaration

```
public Transform LaserHitEffect
```

Field Value

TYPE	DESCRIPTION
Transform	

MaxLength

Declaration

```
public float MaxLength
```

Field Value

TYPE	DESCRIPTION
System.Single	

MaxTime

Declaration

```
public float MaxTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

Offset

Declaration

```
public float Offset
```

Field Value

TYPE	DESCRIPTION
System.Single	

shrinkValue

Declaration

```
public float shrinkValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

Width

Declaration

```
public float Width
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csLookAt

Inheritance

System.Object

csLookAt

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csLookAt : MonoBehaviour
```

Fields

ts

Declaration

```
public Transform ts
```

Field Value

TYPE	DESCRIPTION
Transform	

Class csMaterial_Change

Inheritance

System.Object

csMaterial_Change

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csMaterial_Change : MonoBehaviour
```

Fields

m_inputMaterial

Declaration

```
public Material m_inputMaterial
```

Field Value

TYPE	DESCRIPTION
Material	

m_reduceFactor

Declaration

```
public float m_reduceFactor
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startFactor

Declaration

```
public float m_startFactor
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_timeToReduce

Declaration

```
public float m_timeToReduce
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_timeToStart

Declaration

```
public float m_timeToStart
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

SetTime(Single, Boolean)

Declaration

```
public void SetTime(float _Duration, bool _Loop)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	_Duration	
System.Boolean	_Loop	

Class csMouseOrbit

Inheritance

System.Object
csMouseOrbit

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csMouseOrbit : MonoBehaviour
```

Fields

CameraDist

Declaration

```
public float CameraDist
```

Field Value

TYPE	DESCRIPTION
System.Single	

distance

Declaration

```
public float distance
```

Field Value

TYPE	DESCRIPTION
System.Single	

Target

Declaration

```
public Transform Target
```

Field Value

TYPE	DESCRIPTION
Transform	

xSpeed

Declaration

```
public float xSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMaxLimit

Declaration

```
public float yMaxLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMinLimit

Declaration

```
public float yMinLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

ySpeed

Declaration

```
public float ySpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csMove

Inheritance

System.Object

csMove

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csMove : MonoBehaviour
```

Fields

MoveSpeed

Declaration

```
public float MoveSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMake

Inheritance

System.Object
csObjectMake

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMake : MonoBehaviour
```

Fields

Delay

Declaration

```
public float Delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

MaxRage

Declaration

```
public float MaxRage
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMake2

Inheritance

System.Object

csObjectMake2

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMake2 : MonoBehaviour
```

Fields

isMinusZ

Declaration

```
public bool isMinusZ
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isX

Declaration

```
public bool isX
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isY

Declaration

```
public bool isY
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isZ

Declaration

```
public bool isZ
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_object_MakeCount

Declaration

```
public int m_object_MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_object_MakeDelay

Declaration

```
public float m_object_MakeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_object_MakeRange

Declaration

```
public float m_object_MakeRange
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMake3

Inheritance

System.Object

csObjectMake3

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMake3 : MonoBehaviour
```

Fields

m_destroyTime

Declaration

```
public float m_destroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_makeDelay

Declaration

```
public float m_makeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makePositions

Declaration

```
public Transform[] m_makePositions
```

Field Value

TYPE	DESCRIPTION
Transform[]	

Class csObjectMake4

Inheritance

System.Object

csObjectMake4

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMake4 : MonoBehaviour
```

Fields

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

MakeCount

Declaration

```
public int MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MaxCount

Declaration

```
public float MaxCount
```

Field Value

TYPE	DESCRIPTION
System.Single	

MaxRage

Declaration

```
public float MaxRage
```

Field Value

TYPE	DESCRIPTION
System.Single	

MaxTime

Declaration

```
public float MaxTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMake5

Inheritance

System.Object

csObjectMake5

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMake5 : MonoBehaviour
```

Fields

m_destroyTime

Declaration

```
public float m_destroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_isCrossMake

Declaration

```
public bool m_isCrossMake
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_object_MakeCount

Declaration

```
public float m_object_MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_object_MakeDelay

Declaration

```
public float m_object_MakeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_objectSize

Declaration

```
public float m_objectSize
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_subtractYValue

Declaration

```
public float m_subtractYValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMake6

Inheritance

System.Object

csObjectMake6

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMake6 : MonoBehaviour
```

Fields

m_DestroyTime

Declaration

```
public float m_DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObject

Declaration

```
public GameObject m_gameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_interval

Declaration

```
public float m_interval
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_object_MakeCount

Declaration

```
public int m_object_MakeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_object_MakeDelay

Declaration

```
public float m_object_MakeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_yPos

Declaration

```
public float m_yPos
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMoveDestroy

Inheritance

System.Object

csObjectMoveDestroy

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMoveDestroy : MonoBehaviour
```

Fields

DestroyTime

Declaration

```
public float DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

DestroyTime2

Declaration

```
public float DestroyTime2
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObjectMain

Declaration

```
public GameObject m_gameObjectMain
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_gameObjectTail

Declaration

```
public GameObject m_gameObjectTail
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_hitObject

Declaration

```
public Transform m_hitObject
```

Field Value

TYPE	DESCRIPTION
Transform	

maxLength

Declaration

```
public float maxLength
```

Field Value

TYPE	DESCRIPTION
System.Single	

maxTime

Declaration

```
public float maxTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

MoveSpeed

Declaration

```
public float MoveSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csObjectMoveDestroy2

Inheritance

System.Object

csObjectMoveDestroy2

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csObjectMoveDestroy2 : MonoBehaviour
```

Fields

DestroyTime

Declaration

```
public float DestroyTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_gameObjectMain

Declaration

```
public GameObject m_gameObjectMain
```

Field Value

TYPE	DESCRIPTION
GameObject	

m_gameObjectTail

Declaration

```
public GameObject m_gameObjectTail
```

Field Value

TYPE	DESCRIPTION
GameObject	

maxTime

Declaration

```
public float maxTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

MoveSpeed

Declaration

```
public float MoveSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csParticleStop

Inheritance

System.Object

csParticleStop

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csParticleStop : MonoBehaviour
```

Fields

m_particleMembers

Declaration

```
public ParticleSystem[] m_particleMembers
```

Field Value

TYPE	DESCRIPTION
ParticleSystem[]	

m_stopTime

Declaration

```
public float m_stopTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csRiseDown

Inheritance

System.Object

csRiseDown

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csRiseDown : MonoBehaviour
```

Fields

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_endTime

Declaration

```
public float m_endTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lerpTime

Declaration

```
public float m_lerpTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startTime

Declaration

```
public float m_startTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_y_Value

Declaration

```
public float m_y_Value
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class csRotate

Inheritance

System.Object

csRotate

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csRotate : MonoBehaviour
```

Fields

RotateOffset

Declaration

```
public Vector3 RotateOffset
```

Field Value

TYPE	DESCRIPTION
Vector3	

Class csScaleChange

Inheritance

System.Object

csScaleChange

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class csScaleChange : MonoBehaviour
```

Fields

isScaleUp

Declaration

```
public bool isScaleUp
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_startTime

Declaration

```
public float m_startTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_Weight

Declaration

```
public float m_Weight
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class CustomRampGeneratorEditor

Inheritance

System.Object

CustomRampGeneratorEditor

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CustomRampGeneratorEditor : Editor
```

Methods

OnInspectorGUI()

Declaration

```
public override void OnInspectorGUI()
```

Class Decolorator

Inheritance

System.Object

Decolorator

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Decolorator : MonoBehaviour
```

Fields

computeShader

Declaration

```
public ComputeShader computeShader
```

Field Value

TYPE	DESCRIPTION
ComputeShader	

imageList

Declaration

```
public List<Image> imageList
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

textures

Declaration

```
public List<Texture> textures
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Texture>	

Methods

BlackAndWhite()

Declaration

```
public void BlackAndWhite()
```

Color()

Declaration

```
public void Color()
```

Init(List<Image>)

Declaration

```
public void Init(List<Image> targetImages)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Image>	targetImages	

Class DelayObjectMake

Inheritance

System.Object

[_ObjectMakeBase](#)

DelayObjectMake

Inherited Members

[_ObjectMakeBase.m_makeObjs](#)

[_ObjectMakeBase.m_movePos](#)

[_ObjectMakeBase.GetRandomValue\(Single\)](#)

[_ObjectMakeBase.GetRandomVector\(Vector3\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DelayObjectMake : _ObjectMakeBase
```

Fields

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class DemoScenesNavigation

Inheritance

System.Object

DemoScenesNavigation

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DemoScenesNavigation : MonoBehaviour
```

Class DemoToonVFX

Inheritance

System.Object

DemoToonVFX

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DemoToonVFX : MonoBehaviour
```

Fields

activationTime

Declaration

```
public float[] activationTime
```

Field Value

TYPE	DESCRIPTION
System.Single[]	

animObject

Declaration

```
public Animator animObject
```

Field Value

TYPE	DESCRIPTION
Animator	

cameraPos

Declaration

```
public Vector3 cameraPos
```

Field Value

TYPE	DESCRIPTION
Vector3	

currDistance

Declaration

```
public float currDistance
```

Field Value

TYPE	DESCRIPTION
System.Single	

Holder

Declaration

```
public Transform Holder
```

Field Value

TYPE	DESCRIPTION
Transform	

HueTexture

Declaration

```
public Texture HueTexture
```

Field Value

TYPE	DESCRIPTION
Texture	

Prefabs

Declaration

```
public GameObject[] Prefabs
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

prevDistance

Declaration

```
public float prevDistance
```

Field Value

TYPE	DESCRIPTION
System.Single	

useAnimation

Declaration

```
public bool useAnimation
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

xRotate

Declaration

```
public float xRotate
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMaxLimit

Declaration

```
public float yMaxLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMinLimit

Declaration

```
public float yMinLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

yRotate

Declaration

```
public float yRotate
```

Field Value

TYPE	DESCRIPTION
System.Single	

Struct DemoToonVFX.SVA

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public struct SVA
```

Fields

A

Declaration

```
public float A
```

Field Value

TYPE	DESCRIPTION
System.Single	

S

Declaration

```
public float S
```

Field Value

TYPE	DESCRIPTION
System.Single	

V

Declaration

```
public float V
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class DisableOtherShinsei

Inheritance

System.Object

DisableOtherShinsei

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DisableOtherShinsei : MonoBehaviour
```

Enum DragonPartType

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum DragonPartType
```

Fields

NAME	DESCRIPTION
cabeza	

Struct DtoBugData

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public struct DtoBugData
```

Fields

data

Declaration

```
public DtoDetailBugData data
```

Field Value

TYPE	DESCRIPTION
DtoDetailBugData	

status

Declaration

```
public int status
```

Field Value

TYPE	DESCRIPTION
System.Int32	

success

Declaration

```
public bool success
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Struct DtoDetailBugData

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public struct DtoDetailBugData
```

Fields

link

Declaration

```
public string link
```

Field Value

TYPE	DESCRIPTION
System.String	

Class DummyMainMenuNavigation

Inheritance

System.Object

DummyMainMenuNavigation

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DummyMainMenuNavigation : MonoBehaviour
```


Class EffectsScene

Inheritance

System.Object
EffectsScene

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class EffectsScene : MonoBehaviour
```

Fields

inputLocation

Declaration

```
public static int inputLocation
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_destroyObjects

Declaration

```
public static GameObject[] m_destroyObjects
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

m_effectName

Declaration

```
public Text m_effectName
```

Field Value

TYPE	DESCRIPTION
Text	

m_effects

Declaration

```
public Transform[] m_effects
```

Field Value

TYPE	DESCRIPTION
Transform[]	

Class EGA_EffectSound

Inheritance

System.Object

EGA_EffectSound

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class EGA_EffectSound : MonoBehaviour
```

Fields

maxVolume

Declaration

```
public float maxVolume
```

Field Value

TYPE	DESCRIPTION
System.Single	

minVolume

Declaration

```
public float minVolume
```

Field Value

TYPE	DESCRIPTION
System.Single	

RandomVolume

Declaration

```
public bool RandomVolume
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Repeating

Declaration

```
public bool Repeating
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

RepeatTime

Declaration

```
public float RepeatTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

StartTime

Declaration

```
public float StartTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class EnableDrink

Inheritance

System.Object

EnableDrink

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class EnableDrink : MonoBehaviour
```

Methods

Drink()

Declaration

```
public void Drink()
```

DrinkOff()

Declaration

```
public void DrinkOff()
```

Class EnableWhenEnable

Inheritance

System.Object

EnableWhenEnable

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class EnableWhenEnable : MonoBehaviour
```

Class Extensions

Inheritance

System.Object

Extensions

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class Extensions
```

Methods

SetActiveToggle(GameObject)

Declaration

```
public static void SetActiveToggle(this GameObject target)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	target	

Enum FirebasePetitionType

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum FirebasePetitionType
```

Fields

NAME	DESCRIPTION
SendTurn	
Surrender	
UpdateParty	

Class Followpoints

Inheritance

System.Object

Followpoints

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Followpoints : MonoBehaviour
```


Class Footsteps

Simple behavior that play music according to the animation

Inheritance

System.Object

Footsteps

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Footsteps : MonoBehaviour
```

Fields

CurrentTerrain

Declaration

```
public string CurrentTerrain
```

Field Value

TYPE	DESCRIPTION
System.String	

MyBank

Declaration

```
public AK.Wwise.Bank MyBank
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.Bank	

MyEvent

Declaration

```
public AK.Wwise.Event MyEvent
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.Event	

Methods

DetectSurface()

Declaration

```
public void DetectSurface()
```

PlayFootSound()

Declaration

```
public void PlayFootSound()
```

SetSwitchUsingNames(String)

Declaration

```
public void SetSwitchUsingNames(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

Start()

Declaration

```
public void Start()
```

Class ForwardMovement

Inheritance

System.Object

ForwardMovement

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ForwardMovement : MonoBehaviour
```

Class FowardObjectMake

Inheritance

System.Object

[_ObjectMakeBase](#)

FowardObjectMake

Inherited Members

[_ObjectMakeBase.m_makeObjs](#)

[_ObjectMakeBase.m_movePos](#)

[_ObjectMakeBase.GetRandomValue\(Single\)](#)

[_ObjectMakeBase.GetRandomVector\(Vector3\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FowardObjectMake : _ObjectMakeBase
```

Fields

m_isCrossMake

Declaration

```
public bool m_isCrossMake
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_makeCount

Declaration

```
public float m_makeCount
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeDelay

Declaration

```
public float m_makeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_objectSize

Declaration

```
public float m_objectSize
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_subtractYValue

Declaration

```
public float m_subtractYValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class FreeCam

A simple free camera to be added to a Unity game object.

Keys: wasd / arrows - movement q/e - up/down (local space) r/f - up/down (world space) pageup/pagedown - up/down (world space) hold shift - enable fast movement mode right mouse - enable free look mouse - free look / rotation

Inheritance

System.Object

FreeCam

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FreeCam : MonoBehaviour
```

Fields

fastMovementSpeed

Speed of camera movement when shift is held down,

Declaration

```
public float fastMovementSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

fastZoomSensitivity

Amount to zoom the camera when using the mouse wheel (fast mode).

Declaration

```
public float fastZoomSensitivity
```

Field Value

TYPE	DESCRIPTION
System.Single	

freeLookSensitivity

Sensitivity for free look.

Declaration

```
public float freeLookSensitivity
```

Field Value

TYPE	DESCRIPTION
System.Single	

movementSpeed

Normal speed of camera movement.

Declaration

```
public float movementSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

zoomSensitivity

Amount to zoom the camera when using the mouse wheel.

Declaration

```
public float zoomSensitivity
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

StartLooking()

Enable free looking.

Declaration

```
public void StartLooking()
```

StopLooking()

Disable free looking.

Declaration

```
public void StopLooking()
```

Class FreeFlyCamera

Inheritance

System.Object

FreeFlyCamera

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FreeFlyCamera : MonoBehaviour
```


Class FreeLookCam

Inheritance

System.Object

FreeLookCam

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FreeLookCam : MonoBehaviour
```

Fields

freeLookCamera

Declaration

```
public GameObject freeLookCamera
```

Field Value

TYPE	DESCRIPTION
GameObject	

mainCamera

Declaration

```
public GameObject mainCamera
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

ActivateCamera(Boolean)

Declaration

```
public void ActivateCamera(bool activateFreelook)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	activateFreelook	

Update()

Declaration

```
public void Update()
```

Class FrontAttack

Inheritance

System.Object

FrontAttack

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FrontAttack : MonoBehaviour
```

Fields

anim

Declaration

```
public Animator[] anim
```

Field Value

TYPE	DESCRIPTION
Animator[]	

changeScale

Declaration

```
public bool changeScale
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

craterPrefab

Declaration

```
public GameObject craterPrefab
```

Field Value

TYPE	DESCRIPTION
GameObject	

delay

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

drug

Declaration

```
public float drug
```

Field Value

TYPE	DESCRIPTION
System.Single	

effectWithAnimation

Declaration

```
public bool effectWithAnimation
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

pivot

Declaration

```
public Transform pivot
```

Field Value

TYPE	DESCRIPTION
Transform	

playMeshEffect

Declaration

```
public bool playMeshEffect
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

playPS

Declaration

```
public bool playPS
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

positionOffset

Declaration

```
public float positionOffset
```

Field Value

TYPE	DESCRIPTION
System.Single	

ps

Declaration

```
public ParticleSystem ps
```

Field Value

TYPE	DESCRIPTION
ParticleSystem	

spawnDuration

Declaration

```
public float spawnDuration
```

Field Value

TYPE	DESCRIPTION
System.Single	

spawnRate

Declaration

```
public float spawnRate
```

Field Value

TYPE	DESCRIPTION
System.Single	

speed

Declaration

```
public float speed
```

Field Value

TYPE	DESCRIPTION
System.Single	

startRotation

Declaration

```
public Vector3 startRotation
```

Field Value

TYPE	DESCRIPTION
Vector3	

Methods

MeshEffect()

Declaration

```
public IEnumerator MeshEffect()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

PrepeareAttack(Vector3)

Declaration

```
public void PrepeareAttack(Vector3 targetPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	targetPoint	

StartMove()

Declaration

```
public IEnumerator StartMove()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class FrontMover

Inheritance

System.Object

FrontMover

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FrontMover : MonoBehaviour
```

Fields

drug

Declaration

```
public float drug
```

Field Value

TYPE	DESCRIPTION
System.Single	

effect

Declaration

```
public ParticleSystem effect
```

Field Value

TYPE	DESCRIPTION
ParticleSystem	

pivot

Declaration

```
public Transform pivot
```

Field Value

TYPE	DESCRIPTION
Transform	

repeatingTime

Declaration

```
public float repeatingTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

speed

Declaration

public float speed

Field Value

TYPE	DESCRIPTION
System.Single	

Class GameController

Inheritance

System.Object

GameController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class GameController : MonoBehaviour
```


Class HeadMessages

Inheritance

System.Object

HeadMessages

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class HeadMessages : MonoBehaviour
```

Methods

ShowMessage(String)

Declaration

```
public void ShowMessage(string msg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	msg	

Class HideAfterSeconds

Inheritance

System.Object

HideAfterSeconds

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class HideAfterSeconds : MonoBehaviour
```

Fields

fade

Declaration

```
public bool fade
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

seconds

Declaration

```
public float seconds
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

StartHideCountdown()

Declaration

```
public void StartHideCountdown()
```

Class HS_CameraController

Inheritance

System.Object

HS_CameraController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class HS_CameraController : MonoBehaviour
```

Fields

cameraPos

Declaration

```
public Vector3 cameraPos
```

Field Value

TYPE	DESCRIPTION
Vector3	

collidingLayers

Declaration

```
public LayerMask collidingLayers
```

Field Value

TYPE	DESCRIPTION
LayerMask	

currDistance

Declaration

```
public float currDistance
```

Field Value

TYPE	DESCRIPTION
System.Single	

Holder

Declaration

```
public Transform Holder
```

Field Value

TYPE	DESCRIPTION
Transform	

prevDistance

Declaration

```
public float prevDistance
```

Field Value

TYPE	DESCRIPTION
System.Single	

xRotate

Declaration

```
public float xRotate
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMaxLimit

Declaration

```
public float yMaxLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMinLimit

Declaration

```
public float yMinLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

yRotate

Declaration

```
public float yRotate
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class HS_CameraShaker

Inheritance

System.Object

HS_CameraShaker

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class HS_CameraShaker : MonoBehaviour
```

Fields

amplitude

Declaration

```
public float amplitude
```

Field Value

TYPE	DESCRIPTION
System.Single	

cameraObject

Declaration

```
public Transform cameraObject
```

Field Value

TYPE	DESCRIPTION
Transform	

duration

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

frequency

Declaration

```
public float frequency
```

Field Value

TYPE	DESCRIPTION
System.Single	

timeRemaining

Declaration

public float timeRemaining

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

Shake(Single, Single, Single, Single)

Declaration

public IEnumerable Shake(float amp, float freq, float dur, float wait)
--

Parameters

TYPE	NAME	DESCRIPTION
System.Single	amp	
System.Single	freq	
System.Single	dur	
System.Single	wait	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Interface IBracketsTournament

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public interface IBracketsTournament
```

Properties

CheckTournamentInscription

Declaration

```
CheckTournamentInscription CheckTournamentInscription { get; }
```

Property Value

TYPE	DESCRIPTION
CheckTournamentInscription	

CheckTournamentStateController

Declaration

```
CheckTournamentStateController CheckTournamentStateController { get; }
```

Property Value

TYPE	DESCRIPTION
CheckTournamentStateController	

Methods

SetAlreadyConnection(Boolean)

Declaration

```
void SetAlreadyConnection(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

ShowPanelBracketsView(Boolean)

Declaration

```
void ShowPanelBracketsView(bool active)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	active	

Class InteractWithPlayerController

Inheritance

System.Object

InteractWithPlayerController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class InteractWithPlayerController : MonoBehaviour
```

Fields

interactionsPanel

Declaration

```
public GameObject interactionsPanel
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

ActivatePieMenu()

Declaration

```
public void ActivatePieMenu()
```


Class InteractWithShinsei

Inheritance

System.Object
InteractWithShinsei

Implements

IPointerClickHandler

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class InteractWithShinsei : MonoBehaviour
```

Methods

OnPointerClick(PointerEventData)

Declaration

```
public void OnPointerClick(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

Implements

IPointerClickHandler

Interface ITimer

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public interface ITimer
```

Methods

UpdateTimer(Single, String, Boolean)

Declaration

```
string UpdateTimer(float timeInSeconds, string colorText = null, bool showHour = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	timeInSeconds	
System.String	colorText	
System.Boolean	showHour	

Returns

TYPE	DESCRIPTION
System.String	

Class IUiTrigger

Inheritance

System.Object

IUiTrigger

[UiTrigger](#)

[UViewerTrigger](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class IUiTrigger : MonoBehaviour
```

Fields

OnTriggerEnterEvent

Declaration

```
protected Action<Collider> OnTriggerEnterEvent
```

Field Value

TYPE	DESCRIPTION
Action<Collider>	

OnTriggerExitEvent

Declaration

```
protected Action<Collider> OnTriggerExitEvent
```

Field Value

TYPE	DESCRIPTION
Action<Collider>	

Methods

Update()

Declaration

```
public void Update()
```

Class JPGGenerator

Inheritance

System.Object

JPGGenerator

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class JPGGenerator : MonoBehaviour
```

Fields

_camera

Declaration

```
public Camera _camera
```

Field Value

TYPE	DESCRIPTION
Camera	

characterDatabase

Declaration

```
public CharacterDatabase characterDatabase
```

Field Value

TYPE	DESCRIPTION
CharacterDatabase	

generator

Declaration

```
public Button generator
```

Field Value

TYPE	DESCRIPTION
Button	

loadingPanel

Declaration

```
public GameObject loadingPanel
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

Capture()

Declaration

```
public void Capture()
```

GenerateRandomJPG()

Declaration

```
public void GenerateRandomJPG()
```

GenerateRandomPNG()

Declaration

```
public void GenerateRandomPNG()
```

Class LeaderboardEntry

Inheritance

System.Object

LeaderboardEntry

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LeaderboardEntry : MonoBehaviour
```

Methods

SetEntyValues(String, String, String)

Declaration

```
public void SetEntyValues(string name, string rank, string score)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	
System.String	rank	
System.String	score	

Class LeaderboardEntryResult

Inheritance

System.Object

LeaderboardEntryResult

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LeaderboardEntryResult
```

Fields

DisplayName

Declaration

```
public string DisplayName
```

Field Value

TYPE	DESCRIPTION
System.String	

Position

Declaration

```
public string Position
```

Field Value

TYPE	DESCRIPTION
System.String	

StatValue

Declaration

```
public string StatValue
```

Field Value

TYPE	DESCRIPTION
System.String	

Class LeaderBoardManager

Inheritance

System.Object

LeaderBoardManager

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LeaderBoardManager : MonoBehaviour
```

Methods

Start()

Declaration

```
public void Start()
```


Class LightFlickerEffect

Component which will flicker a linked light while active by changing its intensity between the min and max values given. The flickering can be sharp or smoothed depending on the value of the smoothing parameter.

Just activate / deactivate this component as usual to pause / resume flicker

Inheritance

System.Object

LightFlickerEffect

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LightFlickerEffect : MonoBehaviour
```

Fields

light

Declaration

```
public Light light
```

Field Value

TYPE	DESCRIPTION
Light	

maxIntensity

Declaration

```
public float maxIntensity
```

Field Value

TYPE	DESCRIPTION
System.Single	

minIntensity

Declaration

```
public float minIntensity
```

Field Value

TYPE	DESCRIPTION
System.Single	

smoothing

Declaration

```
public int smoothing
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

Reset()

Reset the randomness and start again. You usually don't need to call this, deactivating/reactivating is usually fine but if you want a strict restart you can do.

Declaration

```
public void Reset()
```

Class LobbyChatView

Inheritance

System.Object
LobbyChatView

Implements

ILobbyChatViewable

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LobbyChatView : MonoBehaviour
```

Properties

OnSendMessage

Declaration

```
public Action<string> OnSendMessage { get; set; }
```

Property Value

TYPE	DESCRIPTION
Action<System.String>	

Methods

Init()

Declaration

```
public void Init()
```

ShowMessage(MessageDto)

Declaration

```
public void ShowMessage(MessageDto messageDto)
```

Parameters

TYPE	NAME	DESCRIPTION
MessageDto	messageDto	

Implements

ILobbyChatViewable

Class LobbyCompanionPanel

Inheritance

System.Object

LobbyCompanionPanel

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LobbyCompanionPanel : MonoBehaviour
```

Class LocalPlayerData

Inheritance

System.Object

LocalPlayerData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LocalPlayerData
```

Fields

challengedPlayer

Declaration

```
public string challengedPlayer
```

Field Value

TYPE	DESCRIPTION
System.String	

characterState

Declaration

```
public CharacterStateEnum characterState
```

Field Value

TYPE	DESCRIPTION
CharacterStateEnum	

currentCharacterStyle

Declaration

```
public Dictionary<PartsOfCharacter, CharacterStyleInfo> currentCharacterStyle
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<PartsOfCharacter, CharacterStyleInfo >	

currentChatMessages

Declaration

```
public List<ChatMessagePayload> currentChatMessages
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<ChatMessagePayload>	

currentMatchId

Declaration

```
public string currentMatchId
```

Field Value

TYPE	DESCRIPTION
System.String	

Deck

Declaration

```
public Deck Deck
```

Field Value

TYPE	DESCRIPTION
Deck	

entityId

Declaration

```
public string entityId
```

Field Value

TYPE	DESCRIPTION
System.String	

entityType

Declaration

```
public string entityType
```

Field Value

TYPE	DESCRIPTION
System.String	

localPlayerNetId

Declaration

```
public ulong localPlayerNetId
```

Field Value

TYPE	DESCRIPTION
System.UInt64	

onPartyChange

Declaration

```
public Action onPartyChange
```

Field Value

TYPE	DESCRIPTION
Action	

playerName

Declaration

```
public string playerName
```

Field Value

TYPE	DESCRIPTION
System.String	

playfabId

Declaration

```
public string playfabId
```

Field Value

TYPE	DESCRIPTION
System.String	

ShinseiCompanion

Declaration

```
public Shinsei ShinseiCompanion
```

Field Value

TYPE	DESCRIPTION
Shinsei	

ShinseiParty

Declaration

```
public List<Shinsei> ShinseiParty
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

ShinseiVault

Declaration

```
public ShinseiVault ShinseiVault
```

Field Value

TYPE	DESCRIPTION
ShinseiVault	

unlockedStyles

Declaration

```
public Dictionary<PartsOfCharacter, UnlockedCharacterStyleInfo> unlockedStyles
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<PartsOfCharacter, UnlockedCharacterStyleInfo >	

Methods

CastCompressedStyleToDictionary(String)

Declaration

```
public Dictionary<PartsOfCharacter, CharacterStyleInfo> CastCompressedStyleToDictionary(string compressedStyle)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	compressedStyle	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<PartsOfCharacter, CharacterStyleInfo >	

CastDictionaryToCompressedStyle()

Declaration

```
public string CastDictionaryToCompressedStyle()
```

Returns

TYPE	DESCRIPTION
System.String	

Class Logout

Inheritance

System.Object

Logout

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Logout : NetworkBehaviour
```

Methods

CallLogout()

Declaration

```
public void CallLogout()
```

Class LookAtAxis

Inheritance

System.Object

LookAtAxis

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LookAtAxis : MonoBehaviour
```

Class LookPlayer

Inheritance

System.Object

LookPlayer

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LookPlayer : MonoBehaviour
```

Struct MatAnimation

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public struct MatAnimation
```

Fields

attachedEdge

Declaration

```
public MatAnimation.AttachedEdge attachedEdge
```

Field Value

TYPE	DESCRIPTION
MatAnimation.AttachedEdge	

image

Declaration

```
public Texture2D image
```

Field Value

TYPE	DESCRIPTION
Texture2D	

matIndex

Declaration

```
public int matIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

targetValue

Declaration

```
public float targetValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

Enum MatAnimation.AttachedEdge

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum AttachedEdge
```

Fields

NAME	DESCRIPTION
x	
y	
z	

Class MatchConfirmedDto

Inheritance

System.Object

MatchConfirmedDto

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MatchConfirmedDto
```

Fields

matchSpawnPoint

Declaration

```
public int matchSpawnPoint
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Struct MatchData

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public struct MatchData
```

Fields

MatchId

Declaration

```
public string MatchId
```

Field Value

TYPE	DESCRIPTION
System.String	

MatchPlayers

Declaration

```
public List<CombatPlayer> MatchPlayers
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<CombatPlayer>	

Class MatchmakingTrigger

Inheritance

System.Object

MatchmakingTrigger

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MatchmakingTrigger : NetworkBehaviour
```


Class MatchState

Inheritance

System.Object
MatchState

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class MatchState
```

Fields

currentTurn

Declaration

```
public int currentTurn
```

Field Value

TYPE	DESCRIPTION
System.Int32	

isOwnerLocal

Declaration

```
public bool isOwnerLocal
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

playersTurn

Declaration

```
public List<ActionCardDto> playersTurn
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ActionCardDto >	

playerWritedLastTurn

Declaration

```
public bool playerWritedLastTurn
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

skipTurnIndex

Declaration

```
public bool skipTurnIndex
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

winnerId

Declaration

```
public string winnerId
```

Field Value

TYPE	DESCRIPTION
System.String	

Class Material_Change

Inheritance

System.Object

Material_Change

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Material_Change : MonoBehaviour
```

Fields

m_inputMaterial

Declaration

```
public Material m_inputMaterial
```

Field Value

TYPE	DESCRIPTION
Material	

m_reduceFactor

Declaration

```
public float m_reduceFactor
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_timeToReduce

Declaration

```
public float m_timeToReduce
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class MaterialOffset

Inheritance

System.Object

MaterialOffset

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MaterialOffset : MonoBehaviour
```

Class MaterialReskin

Inheritance

System.Object

MaterialReskin

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MaterialReskin : MonoBehaviour
```

Fields

skinList

Declaration

```
public List<AtlasTextures> skinList
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< AtlasTextures >	

Methods

ChangePart()

Declaration

```
public void ChangePart()
```

ChangePart(Int32, Int32)

Declaration

```
public void ChangePart(int targetSkin, int targetPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	targetSkin	
System.Int32	targetPosition	

Init()

Declaration

```
public void Init()
```

InitReskin()

Declaration

```
public void InitReskin()
```

NormalMapToUnityFormat(Texture2D)

Declaration

```
public static Texture2D NormalMapToUnityFormat(Texture2D normalMap)
```

Parameters

TYPE	NAME	DESCRIPTION
Texture2D	normalMap	

Returns

TYPE	DESCRIPTION
Texture2D	

Class MessageView

Inheritance

System.Object
MessageView

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class MessageView : MonoBehaviour
```

Fields

possibleIcons

Declaration

```
public List<Sprite> possibleIcons
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

Methods

ShowMessage(String, Int32, Single)

Declaration

```
public void ShowMessage(string message, int icon = -1, float duration = 2F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	
System.Int32	icon	
System.Single	duration	

Class MessageView.Message

Inheritance

System.Object

MessageView.Message

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class Message
```

Fields

icon

Declaration

```
public int icon
```

Field Value

TYPE	DESCRIPTION
System.Int32	

text

Declaration

```
public string text
```

Field Value

TYPE	DESCRIPTION
System.String	

time

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class MMConditionAttribute

Inheritance

System.Object

MMConditionAttribute

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MMConditionAttribute : PropertyAttribute
```

Constructors

MMConditionAttribute(String)

Declaration

```
public MMConditionAttribute(string conditionBoolean)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	conditionBoolean	

MMConditionAttribute(String, Boolean)

Declaration

```
public MMConditionAttribute(string conditionBoolean, bool hideInInspector)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	conditionBoolean	
System.Boolean	hideInInspector	

Fields

ConditionBoolean

Declaration

```
public string ConditionBoolean
```

Field Value

TYPE	DESCRIPTION
System.String	

Hidden

Declaration

```
public bool Hidden
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Class MMEnumConditionAttribute

Inheritance

System.Object

MMEnumConditionAttribute

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MMEnumConditionAttribute : PropertyAttribute
```

Constructors

MMEnumConditionAttribute(String, Int32[])

Declaration

```
public MMEnumConditionAttribute(string conditionBoolean, params int[] enumValues)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	conditionBoolean	
System.Int32[]	enumValues	

Fields

ConditionEnum

Declaration

```
public string ConditionEnum
```

Field Value

TYPE	DESCRIPTION
System.String	

Hidden

Declaration

```
public bool Hidden
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

ContainsBitFlag(Int32)

Declaration

```
public bool ContainsBitFlag(int enumValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	enumValue	

Returns

TYPE	DESCRIPTION
System.Boolean	

Class MMReadOnlyAttribute

Inheritance

System.Object

MMReadOnlyAttribute

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MMReadOnlyAttribute : PropertyAttribute
```

Class MouseOrbit

Inheritance

System.Object

MouseOrbit

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MouseOrbit : MonoBehaviour
```

Fields

CameraDist

Declaration

```
public float CameraDist
```

Field Value

TYPE	DESCRIPTION
System.Single	

distance

Declaration

```
public float distance
```

Field Value

TYPE	DESCRIPTION
System.Single	

Target

Declaration

```
public Transform Target
```

Field Value

TYPE	DESCRIPTION
Transform	

xSpeed

Declaration

```
public float xSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMaxLimit

Declaration

```
public float yMaxLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

yMinLimit

Declaration

```
public float yMinLimit
```

Field Value

TYPE	DESCRIPTION
System.Single	

ySpeed

Declaration

```
public float ySpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class MoveToObject

Inheritance

System.Object

MoveToObject

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MoveToObject : MonoBehaviour
```

Fields

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lerpValue

Declaration

```
public float m_lerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lookValue

Declaration

```
public float m_lookValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_movePos

Declaration

```
public Transform m_movePos
```

Field Value

TYPE	DESCRIPTION
Transform	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class MoveToTag

Inheritance

System.Object

MoveToTag

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MoveToTag : MonoBehaviour
```

Fields

m_durationTime

Declaration

```
public float m_durationTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lerpValue

Declaration

```
public float m_lerpValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_lookValue

Declaration

```
public float m_lookValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_tag

Declaration

```
public string m_tag
```

Field Value

TYPE	DESCRIPTION
System.String	

Class MultiCircleObjectMake

Inheritance

System.Object

[_ObjectMakeBase](#)

MultiCircleObjectMake

Inherited Members

[_ObjectMakeBase.m_makeObjs](#)

[_ObjectMakeBase.m_movePos](#)

[_ObjectMakeBase.GetRandomValue\(Single\)](#)

[_ObjectMakeBase.GetRandomVector\(Vector3\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MultiCircleObjectMake : _ObjectMakeBase
```

Fields

m_interval

Declaration

```
public float m_interval
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_makeCount

Declaration

```
public int m_makeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_makeDelay

Declaration

```
public float m_makeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_startDelay

Declaration

public float m_startDelay

Field Value

TYPE	DESCRIPTION
System.Single	

Class MultiObjectMake

Inheritance

System.Object

[_ObjectMakeBase](#)

MultiObjectMake

Inherited Members

[_ObjectMakeBase.m_makeObjs](#)

[_ObjectMakeBase.m_movePos](#)

[_ObjectMakeBase.GetRandomValue\(Single\)](#)

[_ObjectMakeBase.GetRandomVector\(Vector3\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MultiObjectMake : _ObjectMakeBase
```

Fields

m_makeCount

Declaration

```
public int m_makeCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

m_makeDelay

Declaration

```
public float m_makeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_randomPos

Declaration

```
public Vector3 m_randomPos
```

Field Value

TYPE	DESCRIPTION
Vector3	

m_randomRot

Declaration

```
public Vector3 m_randomRot
```

Field Value

TYPE	DESCRIPTION
Vector3	

m_startDelay

Declaration

```
public float m_startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class NewLeaderBoard

Inheritance

System.Object

NewLeaderBoard

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NewLeaderBoard : MonoBehaviour
```

Methods

ChangeIsRewardWindow(Boolean)

Declaration

```
public void ChangeIsRewardWindow(bool _isRewardWindows)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	_isRewardWindows	

DrawList(List<LeaderboardElement.ElementData>)

Declaration

```
public void DrawList(List<LeaderboardElement.ElementData> elements)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<LeaderboardElement.ElementData>	elements	

GetListOfDivision(Int32)

Declaration

```
public void GetListOfDivision(int divisionIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	divisionIndex	

ProcessDivisionList(PlayFab.ClientModels.GetLeaderboardResult)

Declaration

```
public void ProcessDivisionList(PlayFab.ClientModels.GetLeaderboardResult result)
```

Parameters

TYPE	NAME	DESCRIPTION
PlayFab.ClientModels.GetLeaderboardResult	result	

ProcessVictorieList(PlayFab.ClientModels.GetLeaderboardResult)

Declaration

```
public void ProcessVictorieList(PlayFab.ClientModels.GetLeaderboardResult result)
```

Parameters

TYPE	NAME	DESCRIPTION
PlayFab.ClientModels.GetLeaderboardResult	result	

StatisticRecursiveSearch(PlayFab.ClientModels.GetLeaderboardResult, String, Int32, Action<PlayFab.ClientModels.GetLeaderboardResult>, Action)

Declaration

```
public void StatisticRecursiveSearch(PlayFab.ClientModels.GetLeaderboardResult result, string division, int iterationNumber, Action<PlayFab.ClientModels.GetLeaderboardResult> Callback, Action OnLastIteration = null)
```

Parameters

TYPE	NAME	DESCRIPTION
PlayFab.ClientModels.GetLeaderboardResult	result	
System.String	division	
System.Int32	iterationNumber	
Action<PlayFab.ClientModels.GetLeaderboardResult>	Callback	
Action	OnLastIteration	

Enum NewLeaderBoard.Division

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum Division
```

Fields

NAME	DESCRIPTION
Bronze	
Champion	
Gold	
Silver	

Class NFTGenerator

Inheritance

System.Object

NFTGenerator

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NFTGenerator : MonoBehaviour
```

Fields

characterDatabase

Declaration

```
public CharacterDatabase characterDatabase
```

Field Value

TYPE	DESCRIPTION
CharacterDatabase	

nftsToCreate

Declaration

```
public int nftsToCreate
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

CreateNFTs()

Declaration

```
public void CreateNFTs()
```

Class NFTsModel

Inheritance

System.Object
NFTsModel

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class NFTsModel
```

Properties

NFTsID

Declaration

```
public string NFTsID { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class NodeEnumAttribute

Draw enums correctly within nodes. Without it, enums show up at the wrong positions.

Inheritance

System.Object

NodeEnumAttribute

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NodeEnumAttribute : PropertyAttribute
```

Remarks

Enums with this attribute are not detected by EditorGui.ChangeCheck due to waiting before executing

Class OpenUrl

Inheritance

System.Object

OpenUrl

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class OpenUrl : MonoBehaviour
```

Fields

url

Declaration

```
public string url
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

OpenUrlMethod()

Declaration

```
public void OpenUrlMethod()
```

Class OrderListener

Deprecate system for listen orders, avoiding put a server.

Inheritance

System.Object
OrderListener

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class OrderListener : MonoBehaviour
```

Methods

ChangeShinsei(Int32, Int32, Combat)

Declaration

```
public void ChangeShinsei(int targetShinsei, int playerIndex, Combat playerCombat)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	targetShinsei	
System.Int32	playerIndex	
Combat	playerCombat	

KnowPlayerIndex(OrderMessage, Action<Int32>, Combat)

Declaration

```
public void KnowPlayerIndex(OrderMessage orderMessage, Action<int> onKnowPlayerIndex, Combat playerCombat)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderMessage	orderMessage	
Action<System.Int32>	onKnowPlayerIndex	
Combat	playerCombat	

UpdateOrdersFromNet()

Declaration

```
public void UpdateOrdersFromNet()
```

UpdateTurn(OrderMessage, Int32, Combat)

Declaration

```
public void UpdateTurn(OrderMessage orderMessage, int playerIndex, Combat playerCombat)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderMessage	orderMessage	
System.Int32	playerIndex	
Combat	playerCombat	

Struct OrderMessage

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public struct OrderMessage
```

Fields

data

Declaration

```
public object data
```

Field Value

TYPE	DESCRIPTION
System.Object	

matchId

Declaration

```
public string matchId
```

Field Value

TYPE	DESCRIPTION
System.String	

messageType

Declaration

```
public OrderMessageType messageType
```

Field Value

TYPE	DESCRIPTION
OrderMessageType	

playerPlayfabId

Declaration

```
public string playerPlayfabId
```

Field Value

TYPE	DESCRIPTION
System.String	

Enum OrderMessageType

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum OrderMessageType
```

Fields

NAME	DESCRIPTION
CREATE_MATCH_DATA	
PLAYER_SURRENDER	
SEND_TURN	
SEND_UPDATE_PARTY	

Class OwnBattleTracker

Inheritance

System.Object

OwnBattleTracker

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class OwnBattleTracker : MonoBehaviour
```

Methods

NotifyAttack(Int32)

Declaration

```
public void NotifyAttack(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

TotalTurns(Int32)

Declaration

```
public void TotalTurns(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	

Class OwnBattleTracker.PromediumObject

Inheritance

System.Object

OwnBattleTracker.PromediumObject

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class PromediumObject
```

Fields

TotalDuration

Declaration

```
public List<int> TotalDuration
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

Methods

AddAtLast(Int32)

Declaration

```
public void AddAtLast(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Class PalleteColors

Inheritance

System.Object

PalleteColors

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class PalleteColors
```

Fields

_paletteColor

Declaration

```
public Color32[] _paletteColor
```

Field Value

TYPE	DESCRIPTION
Color32[]	

_partID

Declaration

```
public string _partID
```

Field Value

TYPE	DESCRIPTION
System.String	

Class PartEntity

Inheritance

System.Object

PartEntity

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartEntity : MonoBehaviour
```

Fields

FOLDER_NAME

Declaration

```
public static string FOLDER_NAME
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

BodyPartID

Declaration

```
public string BodyPartID { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

PutEvolution(Boolean)

Declaration

```
public void PutEvolution(bool isOn)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isOn	

Class PartEntityModel

Inheritance

System.Object

PartEntityModel

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartEntityModel
```

Properties

PartRarityModels

Declaration

```
public List<PartRarityModel> PartRarityModels { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< PartRarityModel >	

PartType

Declaration

```
public PartType PartType { get; }
```

Property Value

TYPE	DESCRIPTION
PartType	

Methods

AddEntity(RarityType, PartEntity)

Declaration

```
public void AddEntity(RarityType _rarityType, PartEntity _partEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
RarityType	_rarityType	
PartEntity	_partEntity	

Class ParticleCollisionInstance

Inheritance

System.Object

ParticleCollisionInstance

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ParticleCollisionInstance : MonoBehaviour
```

Fields

DestroyMainEffect

Declaration

```
public bool DestroyMainEffect
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

DestroyTimeDelay

Declaration

```
public float DestroyTimeDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

EffectsOnCollision

Declaration

```
public GameObject[] EffectsOnCollision
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

Offset

Declaration

```
public float Offset
```

Field Value

TYPE	DESCRIPTION
System.Single	

rotationOffset

Declaration

```
public Vector3 rotationOffset
```

Field Value

TYPE	DESCRIPTION
Vector3	

UseFirePointRotation

Declaration

```
public bool UseFirePointRotation
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnlyRotationOffset

Declaration

```
public bool useOnlyRotationOffset
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UseWorldSpacePosition

Declaration

```
public bool UseWorldSpacePosition
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Class PartIdRelation

Inheritance

System.Object

[CharacterStyleRelation](#)

PartIdRelation

Inherited Members

[CharacterStyleRelation.id](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class PartIdRelation : CharacterStyleRelation
```

Fields

name

Declaration

```
public string name
```

Field Value

TYPE	DESCRIPTION
System.String	

slotType

Declaration

```
public PartsOfCharacter slotType
```

Field Value

TYPE	DESCRIPTION
PartsOfCharacter	

sprite

Declaration

```
public Sprite sprite
```

Field Value

TYPE	DESCRIPTION
Sprite	

Class PartIndex

Index of all the current parts of the character

Inheritance

System.Object

PartIndex

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartIndex : SerializedScriptableObject
```

Fields

accessoriesPartNames

Declaration

```
public List<string> accessoriesPartNames
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

bodyPartNames

Declaration

```
public List<string> bodyPartNames
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

earsPartNames

Declaration

```
public List<string> earsPartNames
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

headPartNames

Declaration

```
public List<string> headPartNames
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

SetNames

Declaration

public Dictionary<string, string> SetNames
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

tailPartNames

Declaration

public List<string> tailPartNames

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

Methods

ClearAllLists()

Declaration

public void ClearAllLists()

GenerateRandomShinsei()

Declaration

public string GenerateRandomShinsei()

Returns

TYPE	DESCRIPTION
System.String	

GetPartsDna(String, Int32)

Declaration

public Dictionary<string, long> GetPartsDna(string shinseiDna, int partDigits)
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	

TYPE	NAME	DESCRIPTION
System.Int32	partDigits	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Int64>	

GetShinseiPartTypes(String, Enum)

Declaration

```
public Dictionary<string, string> GetShinseiPartTypes(string shinseiDna, Enum genEnum)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	
Enum	genEnum	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

GetShinseiRarity(String)

Declaration

```
public RarityType GetShinseiRarity(string shinseiDna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	

Returns

TYPE	DESCRIPTION
RarityType	

GetShinseiType(String)

Declaration

```
public CharacterType GetShinseiType(string shinseiDna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	

Returns

TYPE	DESCRIPTION
CharacterType	

ParsePartDNA(Int64)

Declaration

```
public string ParsePartDNA(long dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	dna	

Returns

TYPE	DESCRIPTION
System.String	

ParseShinseiDNA(String, Int32, Int32)

Declaration

```
public string ParseShinseiDNA(string shinseiDna, int colorDigits, int partDigits)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	
System.Int32	colorDigits	
System.Int32	partDigits	

Returns

TYPE	DESCRIPTION
System.String	

PopulateLists(String)

Declaration

```
public void PopulateLists(string entity)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	entity	

SeparateShinseiParts(String, Int32)

Declaration

```
public Dictionary<string, string> SeparateShinseiParts(string shinseiDna, int partDigits)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	
System.Int32	partDigits	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

SetShinseiColor(String)

Declaration

```
public int SetShinseiColor(string dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
System.Int32	

Class PartMultipliers

Inheritance

System.Object

PartMultipliers

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartMultipliers
```

Fields

commonPartMultiplier

Declaration

```
public float commonPartMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

epicPartMultiplier

Declaration

```
public float epicPartMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

legendaryPartMultiplier

Declaration

```
public float legendaryPartMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

rarePartMultiplier

Declaration

```
public float rarePartMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

uncommonPartMultiplier

Declaration

```
public float uncommonPartMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class PartRarityModel

Inheritance

System.Object

PartRarityModel

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartRarityModel
```

Constructors

PartRarityModel(RarityType, List<PartEntity>)

Declaration

```
public PartRarityModel(RarityType _partRarityType, List<PartEntity> _partEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
RarityType	_partRarityType	
System.Collections.Generic.List< PartEntity >	_partEntity	

Properties

PartEntities

Declaration

```
public List<PartEntity> PartEntities { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< PartEntity >	

PartRarityType

Declaration

```
public RarityType PartRarityType { get; }
```

Property Value

TYPE	DESCRIPTION
RarityType	

Methods

AddEntity(PartEntity)

Declaration

```
public void AddEntity(PartEntity _partEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
PartEntity	_partEntity	

RemoveEntity(PartEntity)

Declaration

```
public void RemoveEntity(PartEntity _partEntity)
```

Parameters

TYPE	NAME	DESCRIPTION
PartEntity	_partEntity	

Class PartSelector

Inheritance

System.Object

PartSelector

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartSelector : MonoBehaviour
```

Fields

currentSelectorPart

Declaration

```
public TMP_Text currentSelectorPart
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

leftButton

Declaration

```
public Button leftButton
```

Field Value

TYPE	DESCRIPTION
Button	

rigthButton

Declaration

```
public Button rigthButton
```

Field Value

TYPE	DESCRIPTION
Button	

selectorName

Declaration

```
public TMP_Text selectorName
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Properties

BodyPart

Declaration

```
public PartType BodyPart { get; }
```

Property Value

TYPE	DESCRIPTION
PartType	

Methods

SetBodyPart(PartType)

Declaration

```
public void SetBodyPart(PartType _bodyPart)
```

Parameters

TYPE	NAME	DESCRIPTION
PartType	_bodyPart	

Class PartSlot

Inheritance

System.Object

PartSlot

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartSlot : MonoBehaviour
```

Fields

bodyPart

Declaration

```
public PartType bodyPart
```

Field Value

TYPE	DESCRIPTION
PartType	

FOLDER_NAME

Declaration

```
public static string FOLDER_NAME
```

Field Value

TYPE	DESCRIPTION
System.String	

m_selectedChild

Declaration

```
public PartEntity m_selectedChild
```

Field Value

TYPE	DESCRIPTION
PartEntity	

Properties

ChildrenLength

Declaration

```
public int ChildrenLength { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Childrens

Declaration

```
public PartEntity[] Childrens { get; }
```

Property Value

TYPE	DESCRIPTION
PartEntity[]	

SelectedChildName

Declaration

```
public string SelectedChildName { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ActiveBodyPart(Int32)

Declaration

```
public void ActiveBodyPart(int _index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	_index	

ActiveBodyPartByName(String)

Declaration

```
public void ActiveBodyPartByName(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

GetIndex()

Declaration

```
public int GetIndex()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Initialize()

Declaration

```
public void Initialize()
```


Class PartStat

Inheritance

System.Object

PartStat

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartStat
```

Fields

baseMultipliers

Declaration

```
public BaseMultipliers baseMultipliers
```

Field Value

TYPE	DESCRIPTION
BaseMultipliers	

partMultipliers

Declaration

```
public PartMultipliers partMultipliers
```

Field Value

TYPE	DESCRIPTION
PartMultipliers	

partType

Declaration

```
public CharacterType partType
```

Field Value

TYPE	DESCRIPTION
CharacterType	

typeStatsAndMultipliers

Declaration

```
public TypeStatsAndMultipliers typeStatsAndMultipliers
```

Field Value

TYPE	DESCRIPTION
TypeStatsAndMultipliers	

Enum PartType

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum PartType
```

Fields

NAME	DESCRIPTION
Accessory	
Body	
Ears	
Head	
Tail	

Class PendingVariableNPC

Call all necessary actions to leave the tournament

Inheritance

System.Object

PendingVariableNPC

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PendingVariableNPC : MonoBehaviour
```

Fields

CheckTournamentInscription

Declaration

```
public CheckTournamentInscription CheckTournamentInscription
```

Field Value

TYPE	DESCRIPTION
CheckTournamentInscription	

Methods

LeaveTournament()

Declaration

```
public void LeaveTournament()
```

ShowVendor(Int32)

Change between two npc with different dialogs in the same place

Declaration

```
public void ShowVendor(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	Index of npc

Class PetInteraction

Inheritance

System.Object

PetInteraction

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PetInteraction : MonoBehaviour
```

Fields

layerMaskForStyle

Declaration

```
public LayerMask layerMaskForStyle
```

Field Value

TYPE	DESCRIPTION
LayerMask	

Methods

Hide()

Declaration

```
public void Hide()
```

Init()

Declaration

```
public void Init()
```

PetRotation()

Declaration

```
public void PetRotation()
```

PlayAnimation(Single)

Declaration

```
public void PlayAnimation(float index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	index	

Show()

Declaration

```
public void Show()
```

Class PlayerDataManager

Inheritance

System.Object

PlayerDataManager

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayerDataManager : MonoBehaviour
```

Fields

cardToTest

Declaration

```
public List<int> cardToTest
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

characterStyleController

Declaration

```
public CharacterStyleController characterStyleController
```

Field Value

TYPE	DESCRIPTION
CharacterStyleController	

currentTournamentId

Declaration

```
public string currentTournamentId
```

Field Value

TYPE	DESCRIPTION
System.String	

currentTournamentStage

Declaration

```
public int currentTournamentStage
```

Field Value

TYPE	DESCRIPTION
System.Int32	

endGamePoint

Declaration

```
public Transform endGamePoint
```

Field Value

TYPE	DESCRIPTION
Transform	

isFrenchKeyboardLayout

Declaration

```
public bool isFrenchKeyboardLayout
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isOnTheTournament

Declaration

```
public bool isOnTheTournament
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

localPlayerData

Declaration

```
public LocalPlayerData localPlayerData
```

Field Value

TYPE	DESCRIPTION
LocalPlayerData	

localPlayerGameObject

Declaration

```
public GameObject localPlayerGameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

playerDataReady

Declaration

```
public Action playerDataReady
```

Field Value

TYPE	DESCRIPTION
Action	

Singleton

Declaration

```
public static PlayerDataManager Singleton
```

Field Value

TYPE	DESCRIPTION
PlayerDataManager	

Methods

ButtonChangeShinseis()

Declaration

```
public void ButtonChangeShinseis()
```

CardManagerFill(Dictionary<String, String>)

Declaration

```
public void CardManagerFill(Dictionary<string, string> data)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	data	

CharacterStyleFill(Dictionary<String, String>)

Declaration

```
public void CharacterStyleFill(Dictionary<string, string> data = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	data	

FillPlayerData()

Declaration

```
public Dictionary<string, string> FillPlayerData()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

FillShinseiData()

Declaration

```
public Dictionary<string, string> FillShinseiData()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

GetShinseis(GetUserDataResult)

Declaration

```
public void GetShinseis(GetUserDataResult getUserDataResult)
```

Parameters

TYPE	NAME	DESCRIPTION
GetUserDataResult	getUserDataResult	

InitPlayerData(LoginResult)

Create NewlyPlayerData

Declaration

```
public void InitPlayerData(LoginResult loginResult)
```

Parameters

TYPE	NAME	DESCRIPTION
LoginResult	loginResult	

PlayerShinseisFill(Dictionary<String, String>)

Declaration

```
public void PlayerShinseisFill(Dictionary<string, string> data)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	data	

SetLocalPlayerId(UInt64)

Declaration

```
public void SetLocalPlayerId(ulong data)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt64	data	

UpdateCharacterStyleForAnyReason()

Declaration

```
public void UpdateCharacterStyleForAnyReason()
```

UpdatePlayerData()

Declaration

```
public void UpdatePlayerData()
```

UpdatePlayerName(UpdateUserTitleDisplayNameResult)

Declaration

```
public void UpdatePlayerName(UpdateUserTitleDisplayNameResult displayNameResult)
```

Parameters

TYPE	NAME	DESCRIPTION
UpdateUserTitleDisplayNameResult	displayNameResult	

Events

OnDataObtained

Declaration

```
public event Action OnDataObtained
```

Event Type

TYPE	DESCRIPTION
Action	

Class PlayerIconController

Inheritance

System.Object

PlayerIconController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayerIconController : MonoBehaviour
```

Fields

backpackIcon

Declaration

```
public GameObject backpackIcon
```

Field Value

TYPE	DESCRIPTION
GameObject	

combatIcon

Declaration

```
public GameObject combatIcon
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

ChangeIcon(CharacterStateEnum)

Declaration

```
public void ChangeIcon(CharacterStateEnum characterState)
```

Parameters

TYPE	NAME	DESCRIPTION
CharacterStateEnum	characterState	

Class PlayerUI

Inheritance

System.Object

PlayerUI

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayerUI : MonoBehaviour
```

Fields

accepted

Declaration

```
public bool accepted
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

matchId

Declaration

```
public string matchId
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

CancelMatch(Boolean)

Declaration

```
public void CancelMatch(bool hideSearch = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	hideSearch	

CleanPreviousMatch(String, Action)

Declaration

```
public void CleanPreviousMatch(string matchId, Action callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
Action	callback	

DisplayMatchmakingOptions()

Declaration

```
public void DisplayMatchmakingOptions()
```

HideNameTag(Boolean)

Declaration

```
public void HideNameTag(bool reset = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	reset	

OnOffDisplayName(Boolean)

Declaration

```
public void OnOffDisplayName(bool toggleState)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	toggleState	

OnSpawn(String)

Declaration

```
public void OnSpawn(string _displayName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_displayName	

SearchMatch(Boolean)

Declaration

```
public void SearchMatch(bool initTimer = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	initTimer	

SearchMatchInitTimer()

Declaration

```
public void SearchMatchInitTimer()
```

SetPlayerNameLabel(String)

Declaration

```
public void SetPlayerNameLabel(string newValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newValue	

TestMatchInit()

Declaration

```
public void TestMatchInit()
```

ViewMatch()

Declaration

```
public void ViewMatch()
```

WaitforOpponentToConfirm(GetMatchResult)

Declaration

```
public IEnumerable WaitforOpponentToConfirm(GetMatchResult matchResult)
```

Parameters

TYPE	NAME	DESCRIPTION
GetMatchResult	matchResult	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class PlayFabGetLeaderboard

Inheritance

System.Object
PlayFabGetLeaderboard

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class PlayFabGetLeaderboard : MonoBehaviour
```

Methods

GetLeaderboardEntries(Int32, Int32, String)

Declaration

```
public void GetLeaderboardEntries(int startPosition, int maxResultCount, string leaderboardName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	startPosition	
System.Int32	maxResultCount	
System.String	leaderboardName	

Events

OnSuccess

Declaration

```
public event Action<string> OnSuccess
```

Event Type

TYPE	DESCRIPTION
Action<System.String>	

Class PlayFabGetLeaderboardAroundPlayer

Inheritance

System.Object

PlayFabGetLeaderboardAroundPlayer

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayFabGetLeaderboardAroundPlayer
```

Methods

GetLeaderboardAroundPlayer(String, Int32, String)

Declaration

```
public void GetLeaderboardAroundPlayer(string playerId, int maxResultCount, string leaderboardName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playerId	
System.Int32	maxResultCount	
System.String	leaderboardName	

Events

OnSuccess

Declaration

```
public event Action<string> OnSuccess
```

Event Type

TYPE	DESCRIPTION
Action<System.String>	

Class PlayfabManager

This works as bridge with playfab, all client petitions pass first here

Inheritance

System.Object

PlayfabManager

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayfabManager : MonoBehaviour
```

Fields

debugDataOnServerRequest

Declaration

```
public bool debugDataOnServerRequest
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugMessagePrefix

Declaration

```
public const string debugMessagePrefix = "<color=orange>PlayfabManager battleServer message: </color>"
```

Field Value

TYPE	DESCRIPTION
System.String	

OnCreateTicketFailed

Declaration

```
public UnityEvent OnCreateTicketFailed
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnCreateTicketSuccess

Declaration

```
public UnityEvent<MultiplayerModels.CreateMatchmakingTicketResult> OnCreateTicketSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<MultiplayerModels.CreateMatchmakingTicketResult>	

OnGetCardsStoreFailed

Declaration

public UnityEvent OnGetCardsStoreFailed

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnGetCardsStoreSuccess

Declaration

public UnityEvent<PlayFab.ClientModels.GetCatalogItemsResult> OnGetCardsStoreSuccess
--

Field Value

TYPE	DESCRIPTION
UnityEvent<PlayFab.ClientModels.GetCatalogItemsResult>	

OnLoginFailed

Declaration

public UnityEvent OnLoginFailed

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnLoginSucces

Declaration

public UnityEvent<LoginResult> OnLoginSucces
--

Field Value

TYPE	DESCRIPTION
UnityEvent<LoginResult>	

OnMatchResultFailed

Declaration

public UnityEvent OnMatchResultFailed

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnMatchResultSuccess

Declaration

```
public UnityEvent<MultiplayerModels.GetMatchResult> OnMatchResultSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<MultiplayerModels.GetMatchResult>	

OnPurchaseCardsFailed

Declaration

```
public UnityEvent<PlayFabError> OnPurchaseCardsFailed
```

Field Value

TYPE	DESCRIPTION
UnityEvent<PlayFabError>	

OnPurchaseCardsSuccess

Declaration

```
public UnityEvent<PurchaseItemResult> OnPurchaseCardsSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<PurchaseItemResult>	

OnSignupSuccess

Declaration

```
public UnityEvent<RegisterPlayFabUserResult> OnSignupSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<RegisterPlayFabUserResult>	

OnTicketResultFailed

Declaration

```
public UnityEvent OnTicketResultFailed
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnTicketResultSuccess

Declaration

```
public UnityEvent<MultiplayerModels.GetMatchmakingTicketResult> OnTicketResultSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<MultiplayerModels.GetMatchmakingTicketResult>	

OnUpdateNameSuccess

Declaration

```
public UnityEvent<UpdateUserTitleDisplayNameResult> OnUpdateNameSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<UpdateUserTitleDisplayNameResult>	

OnUserCurrencyGetFailed

Declaration

```
public UnityEvent OnUserCurrencyGetFailed
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnUserCurrencyGetSuccess

Declaration

```
public UnityEvent<PlayFab.ClientModels.GetUserInventoryResult> OnUserCurrencyGetSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent<PlayFab.ClientModels.GetUserInventoryResult>	

Singleton

Declaration

```
public static PlayfabManager Singleton
```

Field Value

TYPE	DESCRIPTION
PlayfabManager	

Methods

AddPlayerCurrency(Int32)

Declaration

```
public void AddPlayerCurrency(int amountToModify)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amountToModify	

BattleServerCheckMatchConfirm(String, Int32, Action<ExecuteFunctionResult>, Action<PlayFabError>)

Declaration

```
public void BattleServerCheckMatchConfirm(string matchId, int spawnPoint, Action<ExecuteFunctionResult> resultCallback, Action<PlayFabError> errorCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
System.Int32	spawnPoint	
Action<ExecuteFunctionResult>	resultCallback	
Action<PlayFabError>	errorCallback	

BattleServerCheckShinseisChoosed(String, Action<ExecuteFunctionResult>)

Declaration

```
public void BattleServerCheckShinseisChoosed(string matchId, Action<ExecuteFunctionResult> resultCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
Action<ExecuteFunctionResult>	resultCallback	

BattleServerCreateMatch(String, Action<ExecuteFunctionResult>)

Declaration

```
public void BattleServerCreateMatch(string matchId, Action<ExecuteFunctionResult> resultCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
Action<ExecuteFunctionResult>	resultCallback	

BattleServerGetMatchState(String, Boolean, Action<ExecuteFunctionResult>, Boolean)

Declaration

```
public void BattleServerGetMatchState(string matchId, bool isViewer = false, Action<ExecuteFunctionResult> resultCallback = null, bool retryEndMatch = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
System.Boolean	isViewer	
Action<ExecuteFunctionResult>	resultCallback	
System.Boolean	retryEndMatch	

BattleServerSelectShinseis(String, CombatPlayer, List<Int32>, Action<ExecuteFunctionResult>, Action<PlayFabError>)

Declaration

```
public void BattleServerSelectShinseis(string matchId, CombatPlayer playerData, List<int> shinseisPlayer, Action<ExecuteFunctionResult> resultCallback, Action<PlayFabError> errorCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
CombatPlayer	playerData	
System.Collections.Generic.List<System.Int32>	shinseisPlayer	
Action<ExecuteFunctionResult>	resultCallback	
Action<PlayFabError>	errorCallback	

BattleServerSendTurn(String, Int32, Action<ExecuteFunctionResult>, Int32)

Declaration

```
public void BattleServerSendTurn(string matchId, int indexOfCard, Action<ExecuteFunctionResult> resultCallback, int counter = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
System.Int32	indexOfCard	
Action<ExecuteFunctionResult>	resultCallback	
System.Int32	counter	

CancelMatchmaking(String, String)

Declaration

```
public void CancelMatchmaking(string queueName, string ticketId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	queueName	
System.String	ticketId	

CreateMatchTicket(String, String, Int32, String)

Create a matchmaking ticket to the specified Queue

Declaration

```
public void CreateMatchTicket(string uniqueId, string playerType, int skillLevel, string queueName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	uniqueId	
System.String	playerType	
System.Int32	skillLevel	
System.String	queueName	

GetLeaderboardAroundPlayer(String, Int32, String)

Declaration

```
public void GetLeaderboardAroundPlayer(string playerId, int maxResultCount, string leaderboardName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playerId	
System.Int32	maxResultCount	

TYPE	NAME	DESCRIPTION
System.String	leaderboardName	

GetLeaderboardEntries(Int32, Int32, String)

Declaration

```
public void GetLeaderboardEntries(int startPosition, int maxResultCount, string leaderboardName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	startPosition	
System.Int32	maxResultCount	
System.String	leaderboardName	

GetLeaderboardEntries(Int32, Int32, String, Action<GetLeaderboardResult>)

Declaration

```
public void GetLeaderboardEntries(int startPosition, int maxResultCount, string leaderboardName,
Action<GetLeaderboardResult> Callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	startPosition	
System.Int32	maxResultCount	
System.String	leaderboardName	
Action<GetLeaderboardResult>	Callback	

GetMatch(String, String)

Gets the created match under the specified ticket id

Declaration

```
public void GetMatch(string matchId, string queueName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
System.String	queueName	

GetPlayerCurrency()

Declaration

```
public void GetPlayerCurrency()
```

GetStatistics(Action<List<StatisticValue>>)

Declaration

```
public void GetStatistics(Action<List<StatisticValue>> actionCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Collections.Generic.List<StatisticValue>>	actionCallback	

GetStoreCards()

Declaration

```
public void GetStoreCards()
```

GetTicketState(String, String, Action<MultiplayerModels.GetMatchmakingTicketResult>)

call this method every 6 to 10 seconds to query the specified ticket state

Declaration

```
public void GetTicketState(string ticketId, string queueName,  
Action<MultiplayerModels.GetMatchmakingTicketResult> result)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	ticketId	
System.String	queueName	
Action<MultiplayerModels.GetMatchmakingTicketResult>	result	

GetUserData(String, List<String>, Action<PlayFab.ClientModels.GetUserDataResult>)

Declaration

```
public void GetUserData(string playfabId, List<string> targetKeys = null,  
Action<PlayFab.ClientModels.GetUserDataResult> OnResult = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playfabId	
System.Collections.Generic.List<System.String>	targetKeys	
Action<PlayFab.ClientModels.GetUserDataResult>	OnResult	

Login(String, String, Action<PlayFabError>, Boolean)

Declaration


```
public void Login(string email, string password, Action<PlayFabError> errorCallback = null, bool isTournamentCreationLogin = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	email	
System.String	password	
Action<PlayFabError>	errorCallback	
System.Boolean	isTournamentCreationLogin	

PurchaseCard(Int32, UInt32)

Declaration

```
public void PurchaseCard(int itemId, uint itemPrice)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	itemId	
System.UInt32	itemPrice	

RemoveUserData(List<String>, Action<UpdateUserDataResult>, Int32, PlayFab.ClientModels.UserDataPermission)

Remove keys of user data

Declaration

```
public void RemoveUserData(List<string> keysToRemove, Action<UpdateUserDataResult> onResult = null, int timesTrying = 0, PlayFab.ClientModels.UserDataPermission permission = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	keysToRemove	
Action<UpdateUserDataResult>	onResult	
System.Int32	timesTrying	
PlayFab.ClientModels.UserDataPermission	permission	

RequestPasswordRecovery(String, Action<String>, Action<PlayFabError>)

Declaration

```
public void RequestPasswordRecovery(string email, Action<string> successCallback = null, Action<PlayFabError> errorCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	email	
Action<System.String>	successCallback	
Action<PlayFabError>	errorCallback	

SetUserData(Dictionary<String, String>, PlayFab.ClientModels.UserDataPermission)

Declaration

```
public void SetUserData(Dictionary<string, string> newData, PlayFab.ClientModels.UserDataPermission permission = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	newData	
PlayFab.ClientModels.UserDataPermission	permission	

SignUp(String, String, String, Action<PlayFabError>)

Register with user email and password

Declaration

```
public void SignUp(string userName, string userEmail, string password, Action<PlayFabError> errorCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	userName	
System.String	userEmail	
System.String	password	
Action<PlayFabError>	errorCallback	

SubtractPlayerCurrency(Int32)

Declaration

```
public void SubtractPlayerCurrency(int amountToModify)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amountToModify	

UpdateDisplayName(String)

Declaration

```
public void UpdateDisplayName(string newName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newName	

UpdatePlayerStatistics(String, Int32)

Declaration

```
public void UpdatePlayerStatistics(string leaderboardName, int score)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	leaderboardName	
System.Int32	score	

UpdateStatistic(String, Int32)

Declaration

```
public void UpdateStatistic(string statisticName, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	statisticName	
System.Int32	value	

Events

OnGetLeaderboardsArroundPlayerSuccess

Declaration

```
public event Action<string> OnGetLeaderboardsArroundPlayerSuccess
```

Event Type

TYPE	DESCRIPTION
Action<System.String>	

OnGetLeaderboardsSuccess

Declaration

```
public event Action<string> OnGetLeaderboardsSuccess
```

Event Type

TYPE	DESCRIPTION
Action<System.String>	

OnSuccess

Declaration

```
public event Action<string> OnSuccess
```

Event Type

TYPE	DESCRIPTION
Action<System.String>	

Class PlayFabUpdatePlayerStatistics

Inheritance

System.Object

PlayFabUpdatePlayerStatistics

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayFabUpdatePlayerStatistics
```

Methods

UpdatePlayerStatistics(String, Int32)

Declaration

```
public void UpdatePlayerStatistics(string leaderboardName, int score)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	leaderboardName	
System.Int32	score	

Class PrefabSpawner

Inheritance

System.Object

PrefabSpawner

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PrefabSpawner : MonoBehaviour
```

Fields

nameOfThePrefab

Declaration

```
public string nameOfThePrefab
```

Field Value

TYPE	DESCRIPTION
System.String	

prefabs

Declaration

```
public GameObject[] prefabs
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

sceneCamera

Declaration

```
public Camera sceneCamera
```

Field Value

TYPE	DESCRIPTION
Camera	

Methods

ChangePrefabIntex(Boolean)

Declaration

```
public void ChangePrefabIntex(bool bo)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bo	

SpawnPrefab()

Declaration

```
public void SpawnPrefab()
```

Class PutBackgroundMusic

Inheritance

System.Object

PutBackgroundMusic

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PutBackgroundMusic : MonoBehaviour
```


Class PutNightModeForVfx

Inheritance

System.Object

PutNightModeForVfx

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PutNightModeForVfx : MonoBehaviour
```

Fields

targetHour

Declaration

```
public float targetHour
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class RampGeneratorTDE

Inheritance

System.Object

RampGeneratorTDE

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RampGeneratorTDE : MonoBehaviour
```

Fields

mode

Declaration

```
public RampGeneratorTDE.Mode mode
```

Field Value

TYPE	DESCRIPTION
RampGeneratorTDE.Mode	

pathForPNG

Declaration

```
public string pathForPNG
```

Field Value

TYPE	DESCRIPTION
System.String	

procedrualGradientRamp

Declaration

```
public Gradient procedrualGradientRamp
```

Field Value

TYPE	DESCRIPTION
Gradient	

renderers

Declaration

```
public Renderer[] renderers
```

Field Value

TYPE	DESCRIPTION
Renderer[]	

Methods

BakeGradient()

Declaration

```
public void BakeGradient()
```

UpdateRampTexture()

Declaration

```
public void UpdateRampTexture()
```

Enum RampGeneratorTDE.Mode

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum Mode
```

Fields

NAME	DESCRIPTION
BakeAndSaveAsTexture	
CreateAtStart	
UpdateEveryFrame	

Class RandomAnim

Inheritance

System.Object

RandomAnim

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RandomAnim : MonoBehaviour
```

Class RankRewardDto

Inheritance

System.Object

RankRewardDto

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class RankRewardDto
```

Fields

Bronze

Declaration

```
public List<RankRewardEntry> Bronze
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< RankRewardEntry >	

Champion

Declaration

```
public List<RankRewardEntry> Champion
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< RankRewardEntry >	

Gold

Declaration

```
public List<RankRewardEntry> Gold
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< RankRewardEntry >	

Silver

Declaration

```
public List<RankRewardEntry> Silver
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< RankRewardEntry >	

Class RankRewardEntry

Inheritance

System.Object

RankRewardEntry

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class RankRewardEntry
```

Fields

position

Declaration

```
public int? position
```

Field Value

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

rewards

Declaration

```
public List<RewardEntry> rewards
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< RewardEntry >	

thresholdDown

Declaration

```
public float thresholdDown
```

Field Value

TYPE	DESCRIPTION
System.Single	

thresholdUp

Declaration

```
public float thresholdUp
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class RareThing

This script solves a problem with Wwise when you put the game in second plane at return all songs in queue play at time, and control the bird sounds

Inheritance

System.Object

RareThing

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RareThing : MonoBehaviour
```

Fields

Birds

Declaration

```
public AK.Wwise.Event Birds
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.Event	

MyEvent

Declaration

```
public AK.Wwise.Event MyEvent
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.Event	

Methods

PlaySound(String)

Declaration

```
public void PlaySound(string state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	state	

Enum RarityType

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum RarityType
```

Fields

NAME	DESCRIPTION
Common	
Epic	
Legendary	
Legendary1	
Legendary2	
Legendary3	
Legendary4	
Legendary5	
Rare	
Uncommon	

Class RecolorBehavior

Inheritance

System.Object

RecolorBehavior

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RecolorBehavior : MonoBehaviour
```

Fields

possibleColors

Declaration

```
public List<Color> possibleColors
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Color>	

Methods

Init()

Declaration

```
public void Init()
```

Class RecolorBtn

Inheritance

System.Object

RecolorBtn

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RecolorBtn : MonoBehaviour
```

Class ReportBug

Inheritance

System.Object

ReportBug

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ReportBug : MonoBehaviour
```

Methods

SendBug()

Declaration

```
public void SendBug()
```

TournamentRequest()

Declaration

```
public IEnumerator TournamentRequest()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class RewardEntry

Inheritance

System.Object

RewardEntry

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class RewardEntry
```

Fields

amount

Declaration

```
public int amount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

type

Declaration

```
public string type
```

Field Value

TYPE	DESCRIPTION
System.String	

Class RewardsRankingSystemController

Inheritance

System.Object

RewardsRankingSystemController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RewardsRankingSystemController : MonoBehaviour
```

Fields

container

Declaration

```
public Transform container
```

Field Value

TYPE	DESCRIPTION
Transform	

prefabEntry

Declaration

```
public EntryController prefabEntry
```

Field Value

TYPE	DESCRIPTION
EntryController	

rowPrefab

Declaration

```
public GameObject rowPrefab
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

InitRewardSystem(Division)

Declaration

```
public void InitRewardSystem(Division rewardSystem)
```

Parameters

TYPE	NAME	DESCRIPTION
Division	rewardSystem	

Class RotateObject

Inheritance

System.Object

RotateObject

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RotateObject : MonoBehaviour
```

Fields

m_delay

Declaration

```
public float m_delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

RotateOffset

Declaration

```
public Vector3 RotateOffset
```

Field Value

TYPE	DESCRIPTION
Vector3	

Class RotateWithDrag

Inheritance

System.Object

RotateWithDrag

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RotateWithDrag : MonoBehaviour
```

Class RotatorPillow

Inheritance

System.Object

RotatorPillow

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RotatorPillow : MonoBehaviour
```

Class ScaleChange

Inheritance

System.Object

ScaleChange

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ScaleChange : MonoBehaviour
```

Fields

m_startTime

Declaration

```
public float m_startTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_Weight

Declaration

```
public float m_Weight
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class ScrollArrows

Inheritance

System.Object

ScrollArrows

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ScrollArrows : MonoBehaviour
```

Class SelectableUi

Inheritance

System.Object
SelectableUi

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class SelectableUi : MonoBehaviour
```

Fields

dontAddOnClickToButtons

Declaration

```
public bool dontAddOnClickToButtons
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

selectable

Declaration

```
public int selectable
```

Field Value

TYPE	DESCRIPTION
System.Int32	

selectableButtons

Declaration

```
public List<SelectableUiButton> selectableButtons
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< SelectableUiButton >	

Methods

Awake()

Declaration

```
public void Awake()
```

InitButtons()

Declaration

```
public void InitButtons()
```

OnClick(SelectableUiButton, Boolean)

Declaration

```
public void OnClick(SelectableUiButton selectableUiButton, bool selected = false)
```

Parameters

TYPE	NAME	DESCRIPTION
SelectableUiButton	selectableUiButton	
System.Boolean	selected	

TryChangeStateToSelectable()

Declaration

```
public void TryChangeStateToSelectable()
```


Class SelectableUIButton

Inheritance

System.Object

SelectableUIButton

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SelectableUIButton : MonoBehaviour
```

Fields

button

Declaration

```
public Button button
```

Field Value

TYPE	DESCRIPTION
Button	

Methods

SetDisable()

Declaration

```
public void SetDisable()
```

SetEnable()

Declaration

```
public void SetEnable()
```

SetSelected()

Declaration

```
public void SetSelected()
```

SetUnselected()

Declaration

```
public void SetUnselected()
```

Class SerializableVector3

Inheritance

System.Object

SerializableVector3

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SerializableVector3
```

Constructors

SerializableVector3(Single, Single, Single)

Declaration

```
public SerializableVector3(float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	x	
System.Single	y	
System.Single	z	

SerializableVector3(Vector3)

Declaration

```
public SerializableVector3(Vector3 vector3)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	vector3	

Fields

x

Declaration

```
public float x
```

Field Value

TYPE	DESCRIPTION
System.Single	

y

Declaration

```
public float y
```

Field Value

TYPE	DESCRIPTION
System.Single	

z

Declaration

public float z

Field Value

TYPE	DESCRIPTION
System.Single	

Class SettingsController

Inheritance

System.Object

SettingsController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SettingsController : MonoBehaviour
```

Methods

ToggleSettingsPopUp()

Declaration

```
public void ToggleSettingsPopUp()
```

Class Shinsei

Inheritance

System.Object

Shinsei

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class Shinsei
```

Fields

alteredStates

Declaration

```
public Dictionary<AlteredStateEnum, AlteredStateData> alteredStates
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary< AlteredStateEnum , AlteredStateData >	

evadeChance

Declaration

```
public int evadeChance
```

Field Value

TYPE	DESCRIPTION
System.Int32	

generation

Declaration

```
public string generation
```

Field Value

TYPE	DESCRIPTION
System.String	

reflectDamage

Declaration

```
public int reflectDamage
```

Field Value

TYPE	DESCRIPTION
System.Int32	

ShinseiActionsIndex

Declaration

public List<int> ShinseiActionsIndex

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

ShinseiDna

Declaration

public string ShinseiDna

Field Value

TYPE	DESCRIPTION
System.String	

shinseiEnergy

Declaration

public int shinseiEnergy

Field Value

TYPE	DESCRIPTION
System.Int32	

shinseiHealth

Declaration

public int shinseiHealth

Field Value

TYPE	DESCRIPTION
System.Int32	

shinseiIcon

Declaration

public Sprite shinseiIcon

Field Value

TYPE	DESCRIPTION
Sprite	

shinseiName

Declaration

public string shinseiName

Field Value

TYPE	DESCRIPTION
System.String	

ShinseiOriginalStats

Declaration

public ShinseiStats ShinseiOriginalStats
--

Field Value

TYPE	DESCRIPTION
ShinseiStats	

shinseiRarity

Declaration

public RarityType shinseiRarity

Field Value

TYPE	DESCRIPTION
RarityType	

shinseiType

Declaration

public CharacterType shinseiType

Field Value

TYPE	DESCRIPTION
CharacterType	

Class ShinseiAnimBehaviour

Inheritance

System.Object

ShinseiAnimBehaviour

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiAnimBehaviour : StateMachineBehaviour
```

Methods

OnStateExit(Animator, AnimatorStateInfo, Int32)

Declaration

```
public override void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
Animator	animator	
AnimatorStateInfo	stateInfo	
System.Int32	layerIndex	

Enum ShinseiStatsEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum ShinseiStatsEnum
```

Fields

NAME	DESCRIPTION
Attack	
Defence	
Speed	
Stamina	
Vigor	

Class ShinseiVault

Inheritance

System.Object
ShinseiVault

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class ShinseiVault
```

Fields

ShinseiVaultList

Declaration

```
public List<Shinsei> ShinseiVaultList
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

Class ShouldSerializeContractResolver

Inheritance

System.Object

ShouldSerializeContractResolver

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShouldSerializeContractResolver : DefaultContractResolver
```

Fields

Instance

Declaration

```
public static readonly ShouldSerializeContractResolver Instance
```

Field Value

TYPE	DESCRIPTION
ShouldSerializeContractResolver	

Methods

CreateProperty(MemberInfo, MemberSerialization)

Declaration

```
protected override JsonProperty CreateProperty(MemberInfo member, MemberSerialization memberSerialization)
```

Parameters

TYPE	NAME	DESCRIPTION
MemberInfo	member	
MemberSerialization	memberSerialization	

Returns

TYPE	DESCRIPTION
JsonProperty	

Class ShowHideDebug

Inheritance

System.Object

ShowHideDebug

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShowHideDebug : MonoBehaviour
```

Class ShowPreviewOfSlot

Inheritance

System.Object

ShowPreviewOfSlot

Implements

IPointerEnterHandler

IPointerExitHandler

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShowPreviewOfSlot : MonoBehaviour
```

Fields

shinseiPreviewPanel

Declaration

```
public ShinseiPreviewPanelManager shinseiPreviewPanel
```

Field Value

TYPE	DESCRIPTION
ShinseiPreviewPanelManager	

Methods

OnPointerEnter(PointerEventData)

Declaration

```
public void OnPointerEnter(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

OnPointerExit(PointerEventData)

Declaration

```
public void OnPointerExit(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

Implements

IPointerEnterHandler

IPointerExitHandler

Class SimpleObjectMake

Inheritance

System.Object

[_ObjectMakeBase](#)

SimpleObjectMake

Inherited Members

[_ObjectMakeBase.m_makeObjs](#)

[_ObjectMakeBase.m_movePos](#)

[_ObjectMakeBase.GetRandomValue\(Single\)](#)

[_ObjectMakeBase.GetRandomVector\(Vector3\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleObjectMake : _ObjectMakeBase
```

Fields

m_randomRotationValue

Declaration

```
public Vector3 m_randomRotationValue
```

Field Value

TYPE	DESCRIPTION
Vector3	

Class SimpleVideoPlayer

Inheritance

System.Object

SimpleVideoPlayer

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleVideoPlayer : MonoBehaviour
```

Class SineCameraControllerTopDownEffects

Inheritance

System.Object

SineCameraControllerTopDownEffects

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SineCameraControllerTopDownEffects : MonoBehaviour
```

Fields

basePivot

Declaration

```
public Transform basePivot
```

Field Value

TYPE	DESCRIPTION
Transform	

camera

Declaration

```
public Camera camera
```

Field Value

TYPE	DESCRIPTION
Camera	

farPivot

Declaration

```
public Transform farPivot
```

Field Value

TYPE	DESCRIPTION
Transform	

maximumAngle

Declaration

```
public float maximumAngle
```

Field Value

TYPE	DESCRIPTION
System.Single	

rotationAmount

Declaration

```
public float rotationAmount
```

Field Value

TYPE	DESCRIPTION
System.Single	

rotationSpeed

Declaration

```
public float rotationSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

scrollSpeed

Declaration

```
public float scrollSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class SineUIControllerTopDownEffects

Inheritance

System.Object
SineUIControllerTopDownEffects

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class SineUIControllerTopDownEffects : MonoBehaviour
```

Fields

canvasGroup

Declaration

```
public CanvasGroup canvasGroup
```

Field Value

TYPE	DESCRIPTION
CanvasGroup	

nameInUI

Declaration

```
public Text nameInUI
```

Field Value

TYPE	DESCRIPTION
Text	

prefabSpawnerObject

Declaration

```
public PrefabSpawner prefabSpawnerObject
```

Field Value

TYPE	DESCRIPTION
PrefabSpawner	

Methods

ChangeEffect(Boolean)

Declaration

```
public void ChangeEffect(bool bo)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bo	

Class SkyChanger

Inheritance

System.Object

SkyChanger

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SkyChanger : MonoBehaviour
```

Methods

ChangeSky(Single)

Declaration

```
public void ChangeSky(float currentTime)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	currentTime	

TestChangeSky()

Declaration

```
public void TestChangeSky()
```

Class StageBracketData

Inheritance

System.Object

StageBracketData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class StageBracketData
```

Fields

brackets

Declaration

```
public Dictionary<string, BracketsData> brackets
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, BracketsData >	

Enum StatValueType

Stat value categories, used of stat calculation

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum StatValueType
```

Fields

NAME	DESCRIPTION
Balance	
Base	
Part	

Class StopMovementWhenEnable

Inheritance

System.Object

StopMovementWhenEnable

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class StopMovementWhenEnable : MonoBehaviour
```

Methods

DisableMovement(Boolean)

Declaration

```
public void DisableMovement(bool newState)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	newState	

Class SwapColorPropertyNameSO

Inheritance

System.Object
SwapColorPropertyNameSO

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class SwapColorPropertyNameSO : ScriptableObject
```

Fields

materialPropertyNames

Declaration

```
public string[] _materialPropertyNames
```

Field Value

TYPE	DESCRIPTION
System.String[]	

Class TargetProjectile

Inheritance

System.Object

TargetProjectile

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TargetProjectile : MonoBehaviour
```

Fields

Detached

Declaration

```
public GameObject[] Detached
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

flash

Declaration

```
public GameObject flash
```

Field Value

TYPE	DESCRIPTION
GameObject	

hit

Declaration

```
public GameObject hit
```

Field Value

TYPE	DESCRIPTION
GameObject	

sideAngle

Declaration

```
public float sideAngle
```

Field Value

TYPE	DESCRIPTION
System.Single	

speed

Declaration

```
public float speed
```

Field Value

TYPE	DESCRIPTION
System.Single	

upAngle

Declaration

```
public float upAngle
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

UpdateTarget(Transform, Vector3)

Declaration

```
public void UpdateTarget(Transform targetPosition, Vector3 Offset)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	targetPosition	
Vector3	Offset	

Class TaskManager

Inheritance

System.Object

TaskManager

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TaskManager : MonoBehaviour
```

Class tasks

Inheritance

System.Object

tasks

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class tasks
```

Fields

tasksList

Declaration

```
public static List<tasks.task> tasksList
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< tasks.task >	

Methods

New(Single, Action)

Declaration

```
public static void New(float requestedTime, Action action)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	requestedTime	
Action	action	

Class tasks.task

Inheritance

System.Object

tasks.task

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class task
```

Fields

action

Declaration

```
public Action action
```

Field Value

TYPE	DESCRIPTION
Action	

startTime

Declaration

```
public float startTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class TerrainChangesDataSO

Inheritance

System.Object
TerrainChangesDataSO

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class TerrainChangesDataSO : SerializedScriptableObject
```

Fields

terrainChanges

Declaration

```
public List<BattleTerrainDataSO> terrainChanges
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleTerrainDataSO >	

Methods

GetJsonAlteredStates()

Declaration

```
public void GetJsonAlteredStates()
```

Class TestExplorer

Inheritance

System.Object

TestExplorer

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TestExplorer : MonoBehaviour
```

Methods

OPEN(String)

Declaration

```
public void OPEN(string itemPath)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	itemPath	

Class ThirdPersonController

Controll the behavior of the main character

Inheritance

System.Object

ThirdPersonController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ThirdPersonController : ThirdPersonCoreController
```

Fields

animator

Declaration

```
public List<Animator> animator
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Animator>	

CanBeBlocked

Declaration

```
public bool CanBeBlocked
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

challengePlayerController

Declaration

```
public ChallengePlayerController challengePlayerController
```

Field Value

TYPE	DESCRIPTION
ChallengePlayerController	

currentMatchId

Declaration

```
public string currentMatchId
```

Field Value

TYPE	DESCRIPTION
System.String	

displayName

Declaration

public string displayName

Field Value

TYPE	DESCRIPTION
System.String	

distance

Declaration

public float distance

Field Value

TYPE	DESCRIPTION
System.Single	

EnableMovement

Declaration

public bool EnableMovement

Field Value

TYPE	DESCRIPTION
System.Boolean	

IsChatMode

Declaration

public bool IsChatMode

Field Value

TYPE	DESCRIPTION
System.Boolean	

IsLocalPlayer

Declaration

public bool IsLocalPlayer

Field Value

TYPE	DESCRIPTION
System.Boolean	

MaxDistanceToTeleport

Declaration

public float MaxDistanceToTeleport

Field Value

TYPE	DESCRIPTION
System.Single	

navmeshAgent

Declaration

public NavMeshAgent navmeshAgent

Field Value

TYPE	DESCRIPTION
NavMeshAgent	

OnDisablePLayer

Declaration

public Action OnDisablePLayer

Field Value

TYPE	DESCRIPTION
Action	

openTournamentButton

Declaration

public Button openTournamentButton

Field Value

TYPE	DESCRIPTION
Button	

playerIconController

Declaration

public PlayerIconController playerIconController
--

Field Value

TYPE	DESCRIPTION
PlayerIconController	

playerPersonalUI

Declaration

public GameObject playerPersonalUI

Field Value

TYPE	DESCRIPTION
GameObject	

playfabId

Declaration

public string playfabId

Field Value

TYPE	DESCRIPTION
System.String	

tournamentReadyController

Declaration

public TournamentReadyController tournamentReadyController
--

Field Value

TYPE	DESCRIPTION
TournamentReadyController	

Properties

IsMovementBloqued

Declaration

public bool IsMovementBloqued { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Init<T>(T)

Declaration

public override void Init<T>(T data)

Parameters

TYPE	NAME	DESCRIPTION
T	data	

Type Parameters

NAME	DESCRIPTION
T	

MoveObject(Nullable<Vector3>)

Declaration

```
public void MoveObject(Vector3? replacePosition = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<Vector3>	replacePosition	

PutPlayerInFacingCamera()

Declaration

```
public void PutPlayerInFacingCamera()
```

SetStateIcon(CharacterStateEnum)

Declaration

```
public void SetStateIcon(CharacterStateEnum characterState)
```

Parameters

TYPE	NAME	DESCRIPTION
CharacterStateEnum	characterState	

Class TimerService

Inheritance

System.Object
TimerService

Implements

[ITimer](#)

IService

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TimerService : MonoBehaviour, ITimer
```

Methods

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

UpdateTimer(Single, String, Boolean)

Declaration

```
public string UpdateTimer(float timeInSeconds, string colorText = null, bool showHour = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	timeInSeconds	
System.String	colorText	
System.Boolean	showHour	

Returns

TYPE	DESCRIPTION
System.String	

Implements

[ITimer](#)

IService

Class TMPChangeColor

Inheritance

System.Object

TMPChangeColor

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TMPChangeColor : MonoBehaviour
```

Fields

currentHighlight

Declaration

```
public Color currentHighlight
```

Field Value

TYPE	DESCRIPTION
Color	

currentNormal

Declaration

```
public Color currentNormal
```

Field Value

TYPE	DESCRIPTION
Color	

highlight

Declaration

```
public Color highlight
```

Field Value

TYPE	DESCRIPTION
Color	

normal

Declaration

```
public Color normal
```

Field Value

TYPE	DESCRIPTION
Color	

Methods

ChangeColor(Boolean)

Declaration

```
public void ChangeColor(bool black)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	black	

Class TournamentBracketsData

Inheritance

System.Object

TournamentBracketsData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentBracketsData
```

Fields

maxPlayers

Declaration

```
public int maxPlayers
```

Field Value

TYPE	DESCRIPTION
System.Int32	

stages

Declaration

```
public Dictionary<string, StageBracketData> stages
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, StageBracketData >	

Class TournamentCamera

Inheritance

System.Object

TournamentCamera

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentCamera : MonoBehaviour
```

Methods

Hide()

Declaration

```
public void Hide()
```

Show()

Declaration

```
public void Show()
```

Class TrackSwitcher

Inheritance

System.Object

TrackSwitcher

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TrackSwitcher : MonoBehaviour
```

Fields

alternativePaths

Declaration

```
public CinemachinePath[] alternativePaths
```

Field Value

TYPE	DESCRIPTION
CinemachinePath[]	

startPath

Declaration

```
public CinemachinePath startPath
```

Field Value

TYPE	DESCRIPTION
CinemachinePath	

Class TranslateMove

Inheritance

System.Object

TranslateMove

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TranslateMove : MonoBehaviour
```

Fields

m_fowardMove

Declaration

```
public bool m_fowardMove
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_power

Declaration

```
public float m_power
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_reduceTime

Declaration

```
public float m_reduceTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_rightMove

Declaration

```
public bool m_rightMove
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

m_upMove

Declaration

```
public bool m_upMove
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Class TransparentBackgroundScreenshotRecorder

Inheritance

System.Object

TransparentBackgroundScreenshotRecorder

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TransparentBackgroundScreenshotRecorder : MonoBehaviour
```

Fields

folderBaseName

Declaration

```
public string folderBaseName
```

Field Value

TYPE	DESCRIPTION
System.String	

frameRate

Declaration

```
public int frameRate
```

Field Value

TYPE	DESCRIPTION
System.Int32	

framesToCapture

Declaration

```
public int framesToCapture
```

Field Value

TYPE	DESCRIPTION
System.Int32	

OnTakePhoto

Declaration

```
public static Action OnTakePhoto
```

Field Value

TYPE	DESCRIPTION
System.Action	

Class TriggerNotificator

Inheritance

System.Object

TriggerNotificator

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TriggerNotificator : MonoBehaviour
```

Methods

Notify()

Declaration

```
public void Notify()
```

Class TypesActions

Inheritance

System.Object

TypesActions

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class TypesActions
```

Fields

actionsData

Declaration

```
public ActionDataList actionsData
```

Field Value

TYPE	DESCRIPTION
ActionDataList	

typeOfShinsei

Declaration

```
public CharacterType typeOfShinsei
```

Field Value

TYPE	DESCRIPTION
CharacterType	

Enum TypesOfTerrainEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum TypesOfTerrainEnum
```

Fields

NAME	DESCRIPTION
Eclipse	
Flames	
Light	
Snow	

Class TypeStatsAndMultipliers

Inheritance

System.Object

TypeStatsAndMultipliers

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TypeStatsAndMultipliers
```

Fields

elementBonusMultiplier1

Declaration

```
public float elementBonusMultiplier1
```

Field Value

TYPE	DESCRIPTION
System.Single	

elementPenaltyMultiplier

Declaration

```
public float elementPenaltyMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

globalPartStat

Declaration

```
public int globalPartStat
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class UIColorSwapper

Inheritance

System.Object

UIColorSwapper

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UIColorSwapper : MonoBehaviour
```

Class UIDisolver

Inheritance

System.Object

UIDisolver

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UIDisolver : MonoBehaviour
```

Methods

SetTargetValue(Single)

Declaration

```
public void SetTargetValue(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	

ToggleMaximumValius()

Declaration

```
public void ToggleMaximumValius()
```

Class UiEmoji

Inheritance

System.Object

UiEmoji

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UiEmoji : MonoBehaviour
```

Methods

Init()

Declaration

```
public void Init()
```

ResetLastEmoji()

Declaration

```
public void ResetLastEmoji()
```

Show(Boolean)

Declaration

```
public void Show(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

Start()

Declaration

```
public void Start()
```

ToggleShow()

Declaration

```
public void ToggleShow()
```

Class UiTrigger

Inheritance

System.Object

[IUiTrigger](#)

UiTrigger

Inherited Members

[IUiTrigger.OnTriggerEnterEvent](#)

[IUiTrigger.OnTriggerExitEvent](#)

[IUiTrigger.Update\(\)](#)

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UiTrigger : IUiTrigger
```

Methods

HideVendor()

Declaration

```
public void HideVendor()
```

ShowVendorUi()

Declaration

```
public void ShowVendorUi()
```

Class UIViewerTrigger

Inheritance

System.Object

IUiTrigger

UIViewerTrigger

Inherited Members

IUiTrigger.OnTriggerEnterEvent

IUiTrigger.OnTriggerExitEvent

IUiTrigger.Update()

Namespace: Global

Assembly: cs.temp.dll.dll

Syntax

```
public class UIViewerTrigger : IUiTrigger
```

Fields

lobbyCamera

Declaration

```
public CinemachineVirtualCamera lobbyCamera
```

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

viewerCamera

Declaration

```
public CinemachineVirtualCamera viewerCamera
```

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

Methods

ChangeCamera()

Declaration

```
public void ChangeCamera()
```

Class UnityNewtonsoftJsonSerializer

Inheritance

System.Object

UnityNewtonsoftJsonSerializer

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UnityNewtonsoftJsonSerializer
```

Methods

Deserialize<T>(String)

Declaration

```
public static T Deserialize<T>(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

Serialize(Object)

Declaration

```
public static string Serialize(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.String	

Class UnlockedCharacterStyleInfo

Inheritance

System.Object
UnlockedCharacterStyleInfo

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class UnlockedCharacterStyleInfo
```

Fields

unlockedColors

Declaration

```
public List<int> unlockedColors
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

unlockedParts

Declaration

```
public List<int> unlockedParts
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

Struct UpdateTurnData

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public struct UpdateTurnData
```

Fields

addToBattleCurrentActions

Declaration

```
public bool addToBattleCurrentActions
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

indexCard

Declaration

```
public int indexCard
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class Vector3Extensions

Inheritance

System.Object

Vector3Extensions

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class Vector3Extensions
```

Methods

FromVector3(Vector3)

Declaration

```
public static SerializableVector3 FromVector3(this Vector3 vector3)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	vector3	

Returns

TYPE	DESCRIPTION
SerializableVector3	

ToVector3(SerializableVector3)

Declaration

```
public static Vector3 ToVector3(this SerializableVector3 serializedVector3)
```

Parameters

TYPE	NAME	DESCRIPTION
SerializableVector3	serializedVector3	

Returns

TYPE	DESCRIPTION
Vector3	

Class VendorHeadTracking

Inheritance

System.Object

VendorHeadTracking

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VendorHeadTracking : MonoBehaviour
```

Methods

SetOriginal()

Declaration

```
public void SetOriginal()
```

SetSeeTarget(Transform)

Declaration

```
public void SetSeeTarget(Transform colisionTarget)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	colisionTarget	

Class VendorSeeArea

Inheritance

System.Object

VendorSeeArea

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VendorSeeArea : NetworkBehaviour
```

Class VersusPanelController

Inheritance

System.Object

VersusPanelController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VersusPanelController : MonoBehaviour
```

Methods

Hide()

Declaration

```
public void Hide()
```

Init(List<Sprite>, List<CharacterType>, String, String)

Declaration

```
public void Init(List<Sprite> sprites, List<CharacterType> types, string playerName, string enemyName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Sprite>	sprites	
System.Collections.Generic.List<CharacterType>	types	
System.String	playerName	
System.String	enemyName	

Show()

Declaration

```
public void Show()
```

Class VFXPositionData

Inheritance

System.Object

VFXPositionData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class VFXPositionData
```

Fields

positions

Declaration

```
public Transform positions
```

Field Value

TYPE	DESCRIPTION
Transform	

vfxPosEnum

Declaration

```
public VFXPositionEnum vfxPosEnum
```

Field Value

TYPE	DESCRIPTION
VFXPositionEnum	

Enum VFXPathPositionEnum

Namespace: [Global](#)
Assembly: cs.temp.dll.dll

Syntax

```
public enum VFXPathPositionEnum
```

Fields

NAME	DESCRIPTION
ARENA_CENTER	
SHINSEI_ENEMY	
SHINSEI_PLAYER	

Class VFXtester

Inheritance

System.Object

VFXtester

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VFXtester : MonoBehaviour
```

Fields

isLocalPlayer

Declaration

```
public bool isLocalPlayer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

targetIndex

Declaration

```
public VFXPositionEnum targetIndex
```

Field Value

TYPE	DESCRIPTION
VFXPositionEnum	

turnsController

Declaration

```
public TurnsController turnsController
```

Field Value

TYPE	DESCRIPTION
TurnsController	

vfxIndex

Declaration

```
public int vfxIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

vFXInstancer

Declaration

```
public VFXInstancer vFXInstancer
```

Field Value

TYPE	DESCRIPTION
VFXInstancer	

Methods

CallVFX()

Declaration

```
public void CallVFX()
```

Class VFXTypeData

Inheritance

System.Object

VFXTypeData

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class VFXTypeData
```

Fields

type

Declaration

```
public CharacterType type
```

Field Value

TYPE	DESCRIPTION
CharacterType	

vfxPrefab

Declaration

```
public GameObject vfxPrefab
```

Field Value

TYPE	DESCRIPTION
GameObject	

Class WhiteMageController

Inheritance

System.Object

WhiteMageController

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WhiteMageController : MonoBehaviour
```

Fields

aim

Declaration

```
public Image aim
```

Field Value

TYPE	DESCRIPTION
Image	

allowPlayerRotation

Declaration

```
public float allowPlayerRotation
```

Field Value

TYPE	DESCRIPTION
System.Single	

anim

Declaration

```
public Animator anim
```

Field Value

TYPE	DESCRIPTION
Animator	

blockRotationPlayer

Declaration

```
public bool blockRotationPlayer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

cam

Declaration

```
public Camera cam
```

Field Value

TYPE	DESCRIPTION
Camera	

cameraShaker

Declaration

```
public HS_CameraShaker cameraShaker
```

Field Value

TYPE	DESCRIPTION
HS_CameraShaker	

canMove

Declaration

```
public bool canMove
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

castingTime

Declaration

```
public float[] castingTime
```

Field Value

TYPE	DESCRIPTION
System.Single[]	

collidingLayer

Declaration

```
public LayerMask collidingLayer
```

Field Value

TYPE	DESCRIPTION
LayerMask	

controller

Declaration

```
public CharacterController controller
```

Field Value

TYPE	DESCRIPTION
CharacterController	

desiredMoveDirection

Declaration

```
public Vector3 desiredMoveDirection
```

Field Value

TYPE	DESCRIPTION
Vector3	

desiredRotationSpeed

Declaration

```
public float desiredRotationSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

FirePoint

Declaration

```
public Transform FirePoint
```

Field Value

TYPE	DESCRIPTION
Transform	

fireRate

Declaration

```
public float fireRate
```

Field Value

TYPE	DESCRIPTION
System.Single	

HorizontalAnimSmoothTime

Declaration

```
public float HorizontalAnimSmoothTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

InputX

Declaration

```
public float InputX
```

Field Value

TYPE	DESCRIPTION
System.Single	

InputZ

Declaration

```
public float InputZ
```

Field Value

TYPE	DESCRIPTION
System.Single	

isGrounded

Declaration

```
public bool isGrounded
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

parentPlace

Declaration

```
public Transform parentPlace
```

Field Value

TYPE	DESCRIPTION
Transform	

Prefabs

Declaration

```
public GameObject[] Prefabs
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

PrefabsCast

Declaration

```
public GameObject[] PrefabsCast
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

screenTargets

Declaration

```
public List<Transform> screenTargets
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Transform>	

skillsRange

Declaration

```
public float skillsRange
```

Field Value

TYPE	DESCRIPTION
System.Single	

Speed

Declaration

```
public float Speed
```

Field Value

TYPE	DESCRIPTION
System.Single	

StartAnimTime

Declaration

```
public float StartAnimTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

StopAnimTime

Declaration

public float StopAnimTime

Field Value

TYPE	DESCRIPTION
System.Single	

TargetMarker

Declaration

public GameObject TargetMarker

Field Value

TYPE	DESCRIPTION
GameObject	

TargetMarker2

Declaration

public GameObject TargetMarker2

Field Value

TYPE	DESCRIPTION
GameObject	

uiOffset

Declaration

public Vector2 uiOffset

Field Value

TYPE	DESCRIPTION
Vector2	

ultIcons

Declaration

public GameObject[] ultIcons

Field Value

TYPE	DESCRIPTION
GameObject[]	

UltimatePrefab

Declaration

public GameObject[] UltimatePrefab

Field Value

TYPE	DESCRIPTION
GameObject[]	

velocity

Declaration

public float velocity

Field Value

TYPE	DESCRIPTION
System.Single	

VerticalAnimTime

Declaration

public float VerticalAnimTime

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

Attack(Int32)

Declaration

public IEnumerator Attack(int EffectNumber)

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

CastSoundPlay()

Declaration

```
public void CastSoundPlay()
```

FastPlay(Int32, Single, Single)

Declaration

```
public IEnumerator FastPlay(int EffectNumber, float castDelay, float endDelay)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	
System.Single	castDelay	
System.Single	endDelay	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

FastPlayTimer(Int32)

Declaration

```
public IEnumerator FastPlayTimer(int EffectNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

FrontAttack(Int32)

Declaration

```
public IEnumerator FrontAttack(int EffectNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

MainSoundPlay()

Declaration

```
public void MainSoundPlay()
```

PreCast(Int32)

Declaration

```
public IEnumerator PreCast(int EffectNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

RotateToTarget(Single, Vector3)

Declaration

```
public IEnumerator RotateToTarget(float rotatingTime, Vector3 targetPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	rotatingTime	
Vector3	targetPoint	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

StopCasting(Int32)

Declaration

```
public void StopCasting(int EffectNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	

targetIndex()

Declaration

```
public int targetIndex()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Ult(Int32, Single, Single, Vector3, Quaternion, Boolean)

Declaration

```
public IEnumerator Ult(int EffectNumber, float enableTime, float dissableTime, Vector3 pivotPosition, Quaternion pivotRotation, bool ChangePos)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	EffectNumber	
System.Single	enableTime	
System.Single	dissableTime	
Vector3	pivotPosition	
Quaternion	pivotRotation	
System.Boolean	ChangePos	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class WorldPositionateElement

Inheritance

System.Object

WorldPositionateElement

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WorldPositionateElement : MonoBehaviour
```

Fields

offset

Declaration

```
public Vector3 offset
```

Field Value

TYPE	DESCRIPTION
Vector3	

target

Declaration

```
public Transform target
```

Field Value

TYPE	DESCRIPTION
Transform	

Class ZoneTrigger

Inheritance

System.Object

ZoneTrigger

Namespace: [Global](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ZoneTrigger : MonoBehaviour
```

Fields

repeatTime

Declaration

```
public float repeatTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

stableEffect

Declaration

```
public ParticleSystem stableEffect
```

Field Value

TYPE	DESCRIPTION
ParticleSystem	

triggerEffect

Declaration

```
public ParticleSystem triggerEffect
```

Field Value

TYPE	DESCRIPTION
ParticleSystem	

Namespace Timba.Characters

Classes

[PartVisual](#)

Class PartVisual

Inheritance

System.Object

PartVisual

Namespace: [Timba.Characters](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartVisual : MonoBehaviour
```

Methods

ChangeSpritePalette(Color32[])

Declaration

```
public void ChangeSpritePalette(Color32[] _colors)
```

Parameters

TYPE	NAME	DESCRIPTION
Color32[]	_colors	

RecolorPartImages(Action<Material>)

Declaration

```
public void RecolorPartImages(Action<Material> _recolorAction)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<Material>	_recolorAction	

RecolorPartSprites(Action<MaterialPropertyBlock>)

Declaration

```
public void RecolorPartSprites(Action<MaterialPropertyBlock> _recolorAction)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<MaterialPropertyBlock>	_recolorAction	

SetPowered(Boolean)

Turns on/off a part. Parts are turned off for building and other no-gameplay situations. Current implementation tries to be generic. Might need to change in the future

Declaration

```
public void SetPowered(bool isPowered)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isPowered	

Events

OnPowered

Declaration

<code>public event Action<bool> OnPowered</code>
--

Event Type

TYPE	DESCRIPTION
Action<System.Boolean>	

Namespace Timba.Games.CharacterFactory

Classes

[CharacterBuilder](#)

Recives a string with the character's structure and Instantiates the respective mesh

[CharacterSelector](#)

[CharacterSlot](#)

[CharacterUtils](#)

[ColorSwapper3D](#)

[PartsStatDatabaseSO](#)

[ShinseiGenerator](#)

Enums

[CharacterType](#)

here should be listed all the possible character families or types to match the assets naming conventions

Class CharacterBuilder

Recives a string with the character's structure and Instantiates the respective mesh

Inheritance

System.Object

Utils.Singleton

Utils.Singleton<[CharacterBuilder](#)>

CharacterBuilder

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterBuilder : Singleton<CharacterBuilder>
```

Methods

InstantiateCharacter(Int32, Transform, Vector3)

Declaration

```
public GameObject InstantiateCharacter(int index, Transform parent, Vector3 position)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	
Transform	parent	
Vector3	position	

Returns

TYPE	DESCRIPTION
GameObject	

UpdateVisual(String, CharacterSlot)

Declaration

```
public void UpdateVisual(string characterStructure, CharacterSlot targetCharacter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	characterStructure	
CharacterSlot	targetCharacter	

Class CharacterSelector

Inheritance

System.Object

CharacterSelector

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterSelector : MonoBehaviour
```

Fields

button

Declaration

```
public Button button
```

Field Value

TYPE	DESCRIPTION
Button	

characterName

Declaration

```
public TMP_Text characterName
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Methods

SetButtonAction(Action, Boolean)

Declaration

```
public void SetButtonAction(Action _action, bool _isClearActionsSubscribed = true)
```

Parameters

TYPE	NAME	DESCRIPTION
Action	_action	
System.Boolean	_isClearActionsSubscribed	

Class CharacterSlot

Inheritance

System.Object
CharacterSlot

Namespace: [Timba.Games.CharacterFactory](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterSlot : MonoBehaviour
```

Fields

animator

Declaration

```
public Animator animator
```

Field Value

TYPE	DESCRIPTION
Animator	

characterCode

Declaration

```
public string characterCode
```

Field Value

TYPE	DESCRIPTION
System.String	

characterID

Declaration

```
public string characterID
```

Field Value

TYPE	DESCRIPTION
System.String	

FOLDER_NAME

Declaration

```
public static string FOLDER_NAME
```

Field Value

TYPE	DESCRIPTION
System.String	

shinsei

Declaration

```
public Shinsei shinsei
```

Field Value

TYPE	DESCRIPTION
Shinsei	

Properties

PartSlots

Declaration

```
public PartSlot[] PartSlots { get; }
```

Property Value

TYPE	DESCRIPTION
PartSlot[]	

Methods

Initialize()

Declaration

```
public void Initialize()
```

SetCharacterCode(String, Boolean)

Declaration

```
public void SetCharacterCode(string code, bool isGenerator = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	code	
System.Boolean	isGenerator	

SetShinseiEvolution(Boolean)

Declaration

```
public void SetShinseiEvolution(bool isOn)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isOn	

UpdateVisual()

Declaration

```
public void UpdateVisual()
```

Enum CharacterType

here should be listed all the possible character families or types to match the assets naming conventions

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public enum CharacterType
```

Fields

NAME	DESCRIPTION
Celestial	
Cursed	
Dark	
Dreamer	
Earth	
Light	
Mecha	
Nature	
Ocean	
Sky	
Snow	
Sun	
Volt	

Class CharacterUtils

Inheritance

System.Object
CharacterUtils

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterUtils : MonoBehaviour
```

Methods

GeneratePartDNA(String)

Declaration

```
public static string GeneratePartDNA(string partEntityName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	partEntityName	

Returns

TYPE	DESCRIPTION
System.String	

GetRarityStat(String)

Declaration

```
public static int GetRarityStat(string raritydna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	raritydna	

Returns

TYPE	DESCRIPTION
System.Int32	

GetTypeValues(Enum, String)

Declaration

```
public static string GetTypeValues(Enum genericEnum, string partName)
```

Parameters

TYPE	NAME	DESCRIPTION
Enum	genericEnum	
System.String	partName	

Returns

TYPE	DESCRIPTION
System.String	

ParsePartDNA(Int64)

Declaration

```
public static string ParsePartDNA(long dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	dna	

Returns

TYPE	DESCRIPTION
System.String	

ParseRarityDNA(Int64)

Declaration

```
public static string ParseRarityDNA(long dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	dna	

Returns

TYPE	DESCRIPTION
System.String	

Class ColorSwapper3D

Inheritance

System.Object

Utils.Singleton

Utils.Singleton<[ColorSwapper3D](#)>

ColorSwapper3D

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ColorSwapper3D : Singleton<ColorSwapper3D>
```

Fields

[_colorsScriptableObject](#)

Declaration

```
public ColorsSO _colorsScriptableObject
```

Field Value

TYPE	DESCRIPTION
ColorsSO	

[_materialColorPropertyName](#)

Declaration

```
public SwapColorPropertyNameSO _materialColorPropertyName
```

Field Value

TYPE	DESCRIPTION
SwapColorPropertyNameSO	

[isCharacterViewScene](#)

Declaration

```
public bool isCharacterViewScene
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

[PaletteID](#)

Declaration

```
public string PaletteID
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

AddTo3DPartList(RecolorablePart3D)

Declaration

```
public void AddTo3DPartList(RecolorablePart3D part)
```

Parameters

TYPE	NAME	DESCRIPTION
RecolorablePart3D	part	

AssignPallet(Int32)

Declaration

```
public void AssignPallet(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

RemovePart(RecolorablePart3D)

Declaration

```
public void RemovePart(RecolorablePart3D part)
```

Parameters

TYPE	NAME	DESCRIPTION
RecolorablePart3D	part	

SwapPalette()

Declaration

```
public void SwapPalette()
```

UpdatePartPalette()

Declaration

```
public void UpdatePartPalette()
```

Class PartsStatDatabaseSO

Inheritance

System.Object

PartsStatDatabaseSO

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PartsStatDatabaseSO : ScriptableObject
```

Fields

partsStats

Declaration

```
public List<PartStat> partsStats
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< PartStat >	

Methods

CalculateStat(PartStat, ShinseiStatsEnum, RarityType, Boolean)

Declaration

```
public int CalculateStat(PartStat partStat, ShinseiStatsEnum bonus, RarityType tier, bool isMainElementStat = false)
```

Parameters

TYPE	NAME	DESCRIPTION
PartStat	partStat	
ShinseiStatsEnum	bonus	
RarityType	tier	
System.Boolean	isMainElementStat	

Returns

TYPE	DESCRIPTION
System.Int32	

GetStat(PartType, String, String, CharacterType, StatValueType)

Declaration

```
public int GetStat(PartType part, string sTier, string sType, CharacterType cType, StatValueType statValueType)
```

Parameters

TYPE	NAME	DESCRIPTION
PartType	part	
System.String	sTier	
System.String	sType	
CharacterType	cType	
StatValueType	statValueType	

Returns

TYPE	DESCRIPTION
System.Int32	

GetStatBalanceValue(ShinseiStatsEnum, CharacterType, RarityType, Int32)

Declaration

```
public int GetStatBalanceValue(ShinseiStatsEnum protectedStat, CharacterType shinseiType, RarityType shinseiTier, int partStatValuesSummatory)
```

Parameters

TYPE	NAME	DESCRIPTION
ShinseiStatsEnum	protectedStat	
CharacterType	shinseiType	
RarityType	shinseiTier	
System.Int32	partStatValuesSummatory	

Returns

TYPE	DESCRIPTION
System.Int32	

Class ShinseiGenerator

Inheritance

System.Object

ShinseiGenerator

Namespace: [Timba.Games.CharacterFactory](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiGenerator : MonoBehaviour
```

Fields

generatedShinseis

Declaration

```
public List<Shinsei> generatedShinseis
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

iconBackground

Declaration

```
public SpriteRenderer iconBackground
```

Field Value

TYPE	DESCRIPTION
SpriteRenderer	

lastIndex

Declaration

```
public int lastIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

photobooth

Declaration

```
public ShinseiWardrobe photobooth
```

Field Value

TYPE	DESCRIPTION
ShinseiWardrobe	

posedShinsei

Declaration

```
public Animator posedShinsei
```

Field Value

TYPE	DESCRIPTION
Animator	

shinseiIcons

Declaration

```
public List<Sprite> shinseiIcons
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

shinseiJsons

Declaration

```
public List<string> shinseiJsons
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

shinseisToGenerate

Declaration

```
public int shinseisToGenerate
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

GenerateShinseis(RarityType, CharacterType)

Declaration

```
public void GenerateShinseis(RarityType desiredTier = RarityType.Common, CharacterType desiredType = CharacterType.Sky)
```

Parameters

TYPE	NAME	DESCRIPTION
RarityType	desiredTier	

TYPE	NAME	DESCRIPTION
CharacterType	desiredType	

GetShinseiNFTData(String, Int32)

Declaration

```
public void GetShinseiNFTData(string shinseiDNA, int shinseiIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDNA	
System.Int32	shinseiIndex	

Namespace Timba.Games.DynamicCamera

Classes

[CameraPlaneController](#)

Switch between diferent cameras by index

Class CameraPlaneController

Switch between diferent cameras by index

Inheritance

System.Object

CameraPlaneController

Namespace: [Timba.Games.DynamicCamera](#)

Assembly: [cs.temp.dll.dll](#)

Syntax

```
public class CameraPlaneController : MonoBehaviour
```

Fields

testStaticCam

Declaration

```
public bool testStaticCam
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

CenitalFrame(Boolean, Action)

Declaration

```
public void CenitalFrame(bool focusPlayer, Action _callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	focusPlayer	
Action	_callback	

ClearPointOfInterest()

Declaration

```
public void ClearPointOfInterest()
```

GeneralFrame(Action)

Declaration

```
public void GeneralFrame(Action _callback)
```

Parameters

TYPE	NAME	DESCRIPTION
Action	_callback	

InitCameras()

Declaration

```
public void InitCameras()
```

InitWaitTurnCameras()

Declaration

```
public void InitWaitTurnCameras()
```

OrbitalCamera(Action)

Declaration

```
public void OrbitalCamera(Action _callback)
```

Parameters

TYPE	NAME	DESCRIPTION
Action	_callback	

StaticCamera(Boolean, Action)

Declaration

```
public void StaticCamera(bool _playerFocus, Action _callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	_playerFocus	
Action	_callback	

StopWaitTurnCameras()

Declaration

```
public void StopWaitTurnCameras()
```

SwitchPointOfInterest(CameraPointOfInteresEnum, Boolean, Boolean)

Declaration

```
public void SwitchPointOfInterest(CameraPointOfInteresEnum pointIndex, bool doFocusOffset = false, bool focusOffsetRight = false)
```

Parameters

TYPE	NAME	DESCRIPTION
CameraPointOfInteresEnum	pointIndex	
System.Boolean	doFocusOffset	
System.Boolean	focusOffsetRight	

SwitchToCam(CamerasAvailableEnum, Single, Single, Action)

Declaration

```
public void SwitchToCam(CamerasAvailableEnum camIndex, float time = 5F, float pathPosition = 0F, Action callback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
CamerasAvailableEnum	camIndex	
System.Single	time	
System.Single	pathPosition	
Action	callback	

WaitForSecondsToCallback(Single, Action)

Declaration

```
public IEnumerator WaitForSecondsToCallback(float seconds, Action callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	seconds	
Action	callback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Namespace Timba.Games.Networking.Testing

Classes

[NetworkingTestUI](#)

Class NetworkingTestUI

Inheritance

System.Object

NetworkingTestUI

Namespace: [Timba.Games.Networking.Testing](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NetworkingTestUI : MonoBehaviour
```

Namespace Timba.Games.Recolor

Classes

[Recolorable](#)

Class Recolorable

Inheritance

System.Object

Recolorable

Namespace: [Timba.Games.Recolor](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Recolorable : MonoBehaviour
```

Methods

SetColors(Color32[])

Declaration

```
public void SetColors(Color32[] newColors)
```

Parameters

TYPE	NAME	DESCRIPTION
Color32[]	newColors	

SetColors(Color32[], Material)

Declaration

```
public void SetColors(Color32[] newColors, Material _materialProperty)
```

Parameters

TYPE	NAME	DESCRIPTION
Color32[]	newColors	
Material	_materialProperty	

SetColors(Color32[], MaterialPropertyBlock)

Declaration

```
public void SetColors(Color32[] newColors, MaterialPropertyBlock _materialProperty)
```

Parameters

TYPE	NAME	DESCRIPTION
Color32[]	newColors	
MaterialPropertyBlock	_materialProperty	

Namespace Timba.Games.SacredTails

Classes

[DisconnectManager](#)

[WalletController](#)

Interfaces

[IWallet](#)

Class DisconnectManager

Inheritance

System.Object

DisconnectManager

Namespace: [Timba.Games.SacredTails](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DisconnectManager : MonoBehaviour
```

Fields

logout

Declaration

```
public Logout logout
```

Field Value

TYPE	DESCRIPTION
Logout	

Methods

LogIn()

Declaration

```
public void LogIn()
```

Update()

Declaration

```
public void Update()
```

Interface IWallet

Namespace: [Timba.Games.SacredTails](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface IWallet
```

Methods

GetUserCurrentCoins()

Declaration

```
int GetUserCurrentCoins()
```

Returns

TYPE	DESCRIPTION
System.Int32	

HideUserWallet()

Declaration

```
void HideUserWallet()
```

ShowUserWallet()

Declaration

```
void ShowUserWallet()
```

UpdateUserWallet()

Declaration

```
void UpdateUserWallet()
```

Class WalletController

Inheritance

System.Object
WalletController

Implements

[IWallet](#)
IService

Namespace: [Timba.Games.SacredTails](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WalletController : MonoBehaviour, IWallet
```

Methods

GetPlayerCurrency(GetUserInventoryResult)

Declaration

```
public void GetPlayerCurrency(GetUserInventoryResult result)
```

Parameters

TYPE	NAME	DESCRIPTION
GetUserInventoryResult	result	

GetCurrentCoins()

Declaration

```
public int GetCurrentCoins()
```

Returns

TYPE	DESCRIPTION
System.Int32	

HideUserWallet()

Declaration

```
public void HideUserWallet()
```

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

ShowUserWallet()

Declaration

```
public void ShowUserWallet()
```

Start()

Declaration

```
public void Start()
```

UpdateUserWallet()

Declaration

```
public void UpdateUserWallet()
```

Implements

[IWallet](#)

IService

Namespace Timba.Games.SacredTails.BattleModule

Classes

[UIHelper](#)

Class UIHelper

Inheritance

System.Object

UIHelper

Namespace: [Timba.Games.SacredTails.BattleModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UIHelper : MonoBehaviour
```

Fields

battleGameMode

Declaration

```
public BattleGameMode battleGameMode
```

Field Value

TYPE	DESCRIPTION
BattleGameMode	

turnsController

Declaration

```
public TurnsController turnsController
```

Field Value

TYPE	DESCRIPTION
TurnsController	

Methods

OpenSkipTurnPopup()

Declaration

```
public void OpenSkipTurnPopup()
```

OpenSurrenderPopup()

Declaration

```
public void OpenSurrenderPopup()
```


Namespace Timba.Games.SacredTails.Lobby

Classes

[PartyManager](#)

[PartyManagerLobby](#)

[PartyManagerVault](#)

[SacredTeleport](#)

Teleport the character from trigger position to targetPosition transform

[ShinseiMovement](#)

This class control the movement of the shinsei companion in the lobby

[ShinseiVaultController](#)

Class PartyManager

Inheritance

System.Object

PartyManager

[PartyManagerLobby](#)

[PartyManagerVault](#)

Namespace: [Timba.Games.SacredTails.Lobby](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class PartyManager : MonoBehaviour
```

Fields

selectorPos

Declaration

```
public int selectorPos
```

Field Value

TYPE	DESCRIPTION
System.Int32	

shinseiSpawner

Declaration

```
public ShinseiSpawner shinseiSpawner
```

Field Value

TYPE	DESCRIPTION
ShinseiSpawner	

Methods

ChangeOnClickSlotAction(UnityAction<Int32, ShinseiSlot>)

Declaration

```
public virtual void ChangeOnClickSlotAction(UnityAction<int, ShinseiSlot> onClick = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityAction<System.Int32, ShinseiSlot>	onClick	

HidePanel()

Declaration

```
public virtual void HidePanel()
```

Initialize(Action<Int32, ShinseiSlot>)

Declaration

```
public virtual void Initialize(Action<int, ShinseiSlot> onNewSlotCreated = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Int32, ShinseiSlot>	onNewSlotCreated	

OnClickSlot(Int32, ShinseiSlot)

Declaration

```
public virtual void OnClickSlot(int listIndex, ShinseiSlot eventShinseiSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	listIndex	
ShinseiSlot	eventShinseiSlot	

Start()

Declaration

```
public virtual void Start()
```

UpdateShinseis()

Declaration

```
public void UpdateShinseis()
```

Class PartyManagerLobby

Inheritance

System.Object

PartyManager

PartyManagerLobby

Inherited Members

PartyManager.selectorPos

PartyManager.shinseiSpawner

PartyManager.Start()

PartyManager.UpdateShinseis()

PartyManager.ChangeOnClickSlotAction(UnityAction<Int32, ShinseiSlot>)

PartyManager.HidePanel()

Namespace: Timba.Games.SacredTails.Lobby

Assembly: cs.temp.dll.dll

Syntax

```
public class PartyManagerLobby : PartyManager
```

Methods

Initialize(Action<Int32, ShinseiSlot>)

Declaration

```
public override void Initialize(Action<int, ShinseiSlot> onNewSlotCreated = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Int32, ShinseiSlot>	onNewSlotCreated	

Overrides

PartyManager.Initialize(Action<Int32, ShinseiSlot>)

OnClickSlot(Int32, ShinseiSlot)

Declaration

```
public override void OnClickSlot(int listIndex, ShinseiSlot eventShinseiSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	listIndex	
ShinseiSlot	eventShinseiSlot	

Overrides

PartyManager.OnClickSlot(Int32, ShinseiSlot)

SwapShinseiBtn(Int32)

Declaration

```
public void SwapShinseiBtn(int dir)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	dir	

Class PartyManagerVault

Inheritance

System.Object

PartyManager

PartyManagerVault

Inherited Members

PartyManager.selectorPos

PartyManager.shinseiSpawner

PartyManager.Start()

PartyManager.UpdateShinseis()

PartyManager.ChangeOnClickSlotAction(UnityAction<Int32, ShinseiSlot>)

PartyManager.HidePanel()

Namespace: Timba.Games.SacredTails.Lobby

Assembly: cs.temp.dll.dll

Syntax

```
public class PartyManagerVault : PartyManager
```

Fields

selectableUi

Declaration

```
public SelectableUi selectableUi
```

Field Value

TYPE	DESCRIPTION
SelectableUi	

shinseiPreviewPanel

Declaration

```
public ShinseiPreviewPanelManager shinseiPreviewPanel
```

Field Value

TYPE	DESCRIPTION
ShinseiPreviewPanelManager	

Methods

ChangeShinseis(Int32, ShinseiSlot, Int32, ShinseiSlot)

Declaration

```
public void ChangeShinseis(int listIndexFrom, ShinseiSlot shinseiSlotFrom, int listIndexTarget, ShinseiSlot shinseiSlotTarget)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	listIndexFrom	
ShinseiSlot	shinseiSlotFrom	
System.Int32	listIndexTarget	
ShinseiSlot	shinseiSlotTarget	

DisplayPreviewForShisnei(Int32)

Declaration

```
public void DisplayPreviewForShisnei(int shinseIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	shinseIndex	

Initialize(Action<Int32, ShinseiSlot>)

Declaration

```
public override void Initialize(Action<int, ShinseiSlot> onNewSlotCreated = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Int32, ShinseiSlot>	onNewSlotCreated	

Overrides

[PartyManager.Initialize\(Action<Int32, ShinseiSlot>\)](#)

OnClickSlot(Int32, ShinseiSlot)

Declaration

```
public override void OnClickSlot(int listIndex, ShinseiSlot eventShinseiSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	listIndex	
ShinseiSlot	eventShinseiSlot	

Overrides

[PartyManager.OnClickSlot\(Int32, ShinseiSlot\)](#)

Class SacredTeleport

Teleport the character from trigger position to targetPosition transform

Inheritance

System.Object

SacredTeleport

Namespace: [Timba.Games.SacredTails.Lobby](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SacredTeleport : MonoBehaviour
```


Class ShinseiMovement

This class control the movement of the shinsei companion in the lobby

Inheritance

System.Object

ShinseiMovement

Namespace: [Timba.Games.SacredTails.Lobby](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiMovement : NetworkBehaviour
```

Fields

navmeshAgent

Declaration

```
public NavMeshAgent navmeshAgent
```

Field Value

TYPE	DESCRIPTION
NavMeshAgent	

Methods

MoveAt(Vector3)

Declaration

```
public void MoveAt(Vector3 targetPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	targetPosition	

SetOwner(Transform)

Declaration

```
public void SetOwner(Transform targetOwner)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	targetOwner	

Class ShinseiVaultController

Inheritance

System.Object

ShinseiVaultController

Namespace: [Timba.Games.SacredTails.Lobby](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiVaultController : MonoBehaviour
```

Methods

DisplayPreviewForShisnei(Int32)

Declaration

```
public void DisplayPreviewForShisnei(int shinseIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	shinseIndex	

Initialize()

Declaration

```
public void Initialize()
```

OnVaultShinseiSelected(Int32, ShinseiSlot)

Declaration

```
public void OnVaultShinseiSelected(int listIndex, ShinseiSlot vaultShinseiSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	listIndex	
ShinseiSlot	vaultShinseiSlot	

Start()

Declaration

```
public void Start()
```

Namespace Timba.Games.SacredTails.Lobby.Chat

Classes

[ChatModule](#)

[Message](#)

Class ChatModule

Inheritance

System.Object
ChatModule

Namespace: [Timba.Games.SacredTails.Lobby.Chat](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChatModule : MonoBehaviour
```

Methods

Init()

Declaration

```
public void Init()
```

SendMessage()

Declaration

```
public void SendMessage()
```

Class Message

Inheritance

System.Object

Message

Namespace: [Timba.Games.SacredTails.Lobby.Chat](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class Message
```

Fields

text

Declaration

```
public string text
```

Field Value

TYPE	DESCRIPTION
System.String	

textObject

Declaration

```
public TMP_Text textObject
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Namespace Timba.Games.SacredTails.LobbyDatabase

Classes

[ChatMessagePayload](#)

[LobbyPlayerBasePayload](#)

[PlayfabLobbyDatabase](#)

[PlayfabTournamentDatabase](#)

[RequestConstants](#)

Structs

[DeletePayload](#)

Interfaces

[ILobbydatabase](#)

[ITournamentDatabase](#)

Enums

[CharacterStateEnum](#)

[TypeOfTournamentRequest](#)

Enum CharacterStateEnum

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public enum CharacterStateEnum
```

Fields

NAME	DESCRIPTION
BACKPACK	
COMBAT	
LOBBY	

Class ChatMessagePayload

Inheritance

System.Object

ChatMessagePayload

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class ChatMessagePayload
```

Fields

id

Declaration

```
public string id
```

Field Value

TYPE	DESCRIPTION
System.String	

message

Declaration

```
public string message
```

Field Value

TYPE	DESCRIPTION
System.String	

timeStamp

Declaration

```
public string timeStamp
```

Field Value

TYPE	DESCRIPTION
System.String	

Struct DeletePayload

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
public struct DeletePayload
```

Fields

encryptIV

Declaration

```
public string encryptIV
```

Field Value

TYPE	DESCRIPTION
System.String	

playfabIdEncrypted

Declaration

```
public string playfabIdEncrypted
```

Field Value

TYPE	DESCRIPTION
System.String	

Interface ILobbydatabase

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface ILobbydatabase
```

Properties

CurrentUserData

Declaration

```
LobbyPlayerBasePayload CurrentUserData { get; }
```

Property Value

TYPE	DESCRIPTION
LobbyPlayerBasePayload	

Methods

ConnectToLobby(Int32, LobbyPlayerBasePayload, Action<ExecuteFunctionResult>)

Declaration

```
void ConnectToLobby(int lobbyId, LobbyPlayerBasePayload userData, Action<ExecuteFunctionResult> resultCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	lobbyId	
LobbyPlayerBasePayload	userData	
Action<ExecuteFunctionResult>	resultCallback	

DisconnectFromLobby(LobbyPlayerBasePayload)

Declaration

```
void DisconnectFromLobby(LobbyPlayerBasePayload userData)
```

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	userData	

RecievePlayersData(Action<Dictionary<String, LobbyPlayerBasePayload>>)

Declaration

```
void RecievePlayersData(Action<Dictionary<string, LobbyPlayerBasePayload>> callback)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Collections.Generic.Dictionary<System.String, LobbyPlayerBasePayload>>	callback	

SendPlayerData(LobbyPlayerBasePayload, Action<Boolean>)

Declaration

```
void SendPlayerData(LobbyPlayerBasePayload userData, Action<bool> onSendSuccess = null)
```

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	userData	
Action<System.Boolean>	onSendSuccess	

SetGetPlayersData(LobbyPlayerBasePayload, Action<ExecuteFunctionResult>)

Declaration

```
void SetGetPlayersData(LobbyPlayerBasePayload userData, Action<ExecuteFunctionResult> resultCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	userData	
Action<ExecuteFunctionResult>	resultCallback	

Interface ITournamentDatabase

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface ITournamentDatabase
```

Methods

TournamentRequest(String, TypeOfTournamentRequest, Action<ExecuteFunctionResult>, Boolean, Int32)

Declaration

```
void TournamentRequest(string tournamentId, TypeOfTournamentRequest typeOfRequest,
Action<ExecuteFunctionResult> resultCallback, bool shouldMarkReady = true, int timesTrying = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	tournamentId	
TypeOfTournamentRequest	typeOfRequest	
Action<ExecuteFunctionResult>	resultCallback	
System.Boolean	shouldMarkReady	
System.Int32	timesTrying	

Class LobbyPlayerBasePayload

Inheritance

System.Object

LobbyPlayerBasePayload

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class LobbyPlayerBasePayload
```

Fields

challengedPlayer

Declaration

```
public string challengedPlayer
```

Field Value

TYPE	DESCRIPTION
System.String	

characterState

Declaration

```
public int characterState
```

Field Value

TYPE	DESCRIPTION
System.Int32	

characterStyle

Declaration

```
public string characterStyle
```

Field Value

TYPE	DESCRIPTION
System.String	

chatMessages

Declaration

```
public List<ChatMessagePayload> chatMessages
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ChatMessagePayload >	

connectionId

Declaration

public ulong connectionId

Field Value

TYPE	DESCRIPTION
System.UInt64	

currentMatchId

Declaration

public string currentMatchId

Field Value

TYPE	DESCRIPTION
System.String	

displayName

Declaration

public string displayName

Field Value

TYPE	DESCRIPTION
System.String	

encryptIV

Declaration

public string encryptIV

Field Value

TYPE	DESCRIPTION
System.String	

lobby

Declaration

public int lobby

Field Value

TYPE	DESCRIPTION
System.Int32	

playerPosition

Declaration

```
public SerializableVector3 playerPosition
```

Field Value

TYPE	DESCRIPTION
SerializableVector3	

playfabIdEncrypted

Declaration

```
public string playfabIdEncrypted
```

Field Value

TYPE	DESCRIPTION
System.String	

shinseiCompanionDna

Declaration

```
public string shinseiCompanionDna
```

Field Value

TYPE	DESCRIPTION
System.String	

Class PlayfabLobbyDatabase

Inheritance

System.Object
PlayfabLobbyDatabase

Implements

[ILobbydatabase](#)
IService

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class PlayfabLobbyDatabase : MonoBehaviour, ILobbydatabase
```

Fields

currentLobby

Declaration

```
public int currentLobby
```

Field Value

TYPE	DESCRIPTION
System.Int32	

currentUserData

Declaration

```
public LobbyPlayerBasePayload currentUserData
```

Field Value

TYPE	DESCRIPTION
LobbyPlayerBasePayload	

testPlayfabId

Declaration

```
public string testPlayfabId
```

Field Value

TYPE	DESCRIPTION
System.String	

testUserData

Declaration

```
public LobbyPlayerBasePayload testUserData
```

Field Value

TYPE	DESCRIPTION
LobbyPlayerBasePayload	

Properties

CurrentUserData

Declaration

```
public LobbyPlayerBasePayload CurrentUserData { get; }
```

Property Value

TYPE	DESCRIPTION
LobbyPlayerBasePayload	

Methods

Awake()

Declaration

```
public void Awake()
```

ConnectToLobby(Int32, LobbyPlayerBasePayload, Action<ExecuteFunctionResult>)

Declaration

```
public void ConnectToLobby(int lobbyId, LobbyPlayerBasePayload userData, Action<ExecuteFunctionResult> resultCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	lobbyId	
LobbyPlayerBasePayload	userData	
Action<ExecuteFunctionResult>	resultCallback	

DisconnectFromLobby(LobbyPlayerBasePayload)

Declaration

```
public void DisconnectFromLobby(LobbyPlayerBasePayload userData)
```

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	userData	

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

RecievePlayersData(Action<Dictionary<String, LobbyPlayerBasePayload>>)

Declaration

<pre>public void RecievePlayersData(Action<Dictionary<string, LobbyPlayerBasePayload>> callback)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Collections.Generic.Dictionary<System.String, LobbyPlayerBasePayload >>	callback	

SendPlayerData(LobbyPlayerBasePayload, Action<Boolean>)

Declaration

<pre>public void SendPlayerData(LobbyPlayerBasePayload userData, Action<bool> onSendSuccess = null)</pre>

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	userData	
Action<System.Boolean>	onSendSuccess	

SetGetPlayersData(LobbyPlayerBasePayload, Action<ExecuteFunctionResult>)

Declaration

<pre>public void SetGetPlayersData(LobbyPlayerBasePayload userData, Action<ExecuteFunctionResult> callback)</pre>

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	userData	
Action<ExecuteFunctionResult>	callback	

Implements

[ILobbydatabase](#)

IService

Class PlayfabTournamentDatabase

Inheritance

System.Object
PlayfabTournamentDatabase

Implements

[ITournamentDatabase](#)
IService

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class PlayfabTournamentDatabase : MonoBehaviour, ITournamentDatabase
```

Fields

debugError

Declaration

```
public bool debugError
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugRequest

Declaration

```
public bool debugRequest
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugResult

Declaration

```
public bool debugResult
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

TournamentRequest(String, TypeOfTournamentRequest, Action<ExecuteFunctionResult>, Boolean, Int32)

Declaration

```
public void TournamentRequest(string tournamentId, TypeOfTournamentRequest typeOfRequest,
Action<ExecuteFunctionResult> resultCallback, bool shouldMarkReady = false, int timesTrying = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	tournamentId	
TypeOfTournamentRequest	typeOfRequest	
Action<ExecuteFunctionResult>	resultCallback	
System.Boolean	shouldMarkReady	
System.Int32	timesTrying	

Implements

[ITournamentDatabase](#)

IService

Class RequestConstants

Inheritance

System.Object

RequestConstants

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: [cs.temp.dll.dll](#)

Syntax

```
public static class RequestConstants
```

Fields

baseUrl

Declaration

```
public const string baseUrl = "https://sacredtails.azurewebsites.net/api/"
```

Field Value

TYPE	DESCRIPTION
System.String	

firebaseError

Declaration

```
public const string firebaseError = "<color=red>Firebase Lobby Database: </color>"
```

Field Value

TYPE	DESCRIPTION
System.String	

firebaseMessage

Declaration

```
public const string firebaseMessage = "<color=blue>Firebase Lobby Database: </color>"
```

Field Value

TYPE	DESCRIPTION
System.String	

Enum TypeOfTournamentRequest

Namespace: [Timba.Games.SacredTails.LobbyDatabase](#)

Assembly: cs.temp.dll.dll

Syntax

```
public enum TypeOfTournamentRequest
```

Fields

NAME	DESCRIPTION
CHECK_READY_STATE	
CHECK_TOURNAMENT_STATE	
GET_CURRENT_BRACKETS_DATA	
GET_TOURNAMENT_LIST	
JOIN_TOURNAMENT	

Namespace Timba.Games.SacredTails.LobbyNetworking

Classes

[ConnectionResponse](#)

[LobbyNetworkingController](#)

[LobbyNetworkingController.DirectMessage](#)

[LobbyNetworkingController.MessageObject](#)

[PlayersStreamData](#)

Interfaces

[ILobbyNetworkManager](#)

Class ConnectionResponse

Inheritance

System.Object

ConnectionResponse

Namespace: [Timba.Games.SacredTails.LobbyNetworking](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ConnectionResponse
```

Fields

connectionId

Declaration

```
public ulong connectionId
```

Field Value

TYPE	DESCRIPTION
System.UInt64	

currentVersion

Declaration

```
public string currentVersion
```

Field Value

TYPE	DESCRIPTION
System.String	

lobbyId

Declaration

```
public int lobbyId
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Interface ILobbyNetworkManager

Namespace: [Timba.Games.SacredTails.LobbyNetworking](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface ILobbyNetworkManager
```

Properties

CurrentPlayer

Declaration

```
ThirdPersonController CurrentPlayer { get; set; }
```

Property Value

TYPE	DESCRIPTION
ThirdPersonController	

CurrentSecondsBetweenPetition

Declaration

```
float CurrentSecondsBetweenPetition { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

OnConnected

Declaration

```
Action<ThirdPersonController> OnConnected { get; set; }
```

Property Value

TYPE	DESCRIPTION
Action< ThirdPersonController >	

Methods

CheckIfOtherPlayerHasChallengeForPlayfabId(String)

Declaration

```
bool CheckIf0therPlayerHasChallengeForPlayfabId(string playfabId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playfabId	

Returns

TYPE	DESCRIPTION
System.Boolean	

CheckIfPlayerHasChallengeOrIsChallenging(String)

Declaration

bool CheckIfPlayerHasChallengeOrIsChallenging(string playfabId)

Parameters

TYPE	NAME	DESCRIPTION
System.String	playfabId	

Returns

TYPE	DESCRIPTION
System.Boolean	

ConnectToLobby(Int32)

Declaration

void ConnectToLobby(int lobbyId)

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	lobbyId	

GetPlayerState(String)

Declaration

CharacterStateEnum GetPlayerState(string playerId)
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	playerId	

Returns

TYPE	DESCRIPTION
CharacterStateEnum	

ManageLobbyPlayers(Dictionary<String, LobbyPlayerBasePayload>)

Declaration

void ManageLobbyPlayers(Dictionary<string, LobbyPlayerBasePayload> usersData)

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, LobbyPlayerBasePayload>	usersData	

ShowPlayerPersonalUI(Boolean)

Declaration

```
void ShowPlayerPersonalUI(bool show = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	show	

TickCheckActivates()

Declaration

```
void TickCheckActivates()
```

ToggleBattleMode(Boolean)

Declaration

```
void ToggleBattleMode(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

Class LobbyNetworkingController

Inheritance

System.Object
LobbyNetworkingController

Implements

[ILobbyNetworkManager](#)
IService

Namespace: [Timba.Games.SacredTails.LobbyNetworking](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class LobbyNetworkingController : MonoBehaviour, ILobbyNetworkManager
```

Fields

chatTextBox

Declaration

```
public ChatTextBox chatTextBox
```

Field Value

TYPE	DESCRIPTION
ChatTextBox	

connected

Declaration

```
public bool connected
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

connectionId

Declaration

```
public ulong connectionId
```

Field Value

TYPE	DESCRIPTION
System.UInt64	

currentPlayer

Declaration

```
public ThirdPersonController currentPlayer
```

Field Value

TYPE	DESCRIPTION
ThirdPersonController	

currentPlayersAvatar

Declaration

public Dictionary<string, ThirdPersonController > currentPlayersAvatar
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, ThirdPersonController >	

currentPlayersData

Declaration

public Dictionary<string, LobbyPlayerBasePayload> currentPlayersData
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, LobbyPlayerBasePayload>	

lerpDuration

Declaration

public float lerpDuration

Field Value

TYPE	DESCRIPTION
System.Single	

loadingScreen

Declaration

public GameObject loadingScreen

Field Value

TYPE	DESCRIPTION
GameObject	

lobbyDatabase

Declaration

public ILobbydatabase lobbyDatabase

Field Value

TYPE	DESCRIPTION
ILobbydatabase	

localPlayerPrefab

Declaration

public ThirdPersonController localPlayerPrefab
--

Field Value

TYPE	DESCRIPTION
ThirdPersonController	

minErrorPosition

Declaration

public float minErrorPosition

Field Value

TYPE	DESCRIPTION
System.Single	

onBattle

Declaration

public bool onBattle

Field Value

TYPE	DESCRIPTION
System.Boolean	

OnConnectedCallback

Declaration

public UnityEvent<ThirdPersonController> OnConnectedCallback
--

Field Value

TYPE	DESCRIPTION
UnityEvent< ThirdPersonController >	

otherPlayerPrefab

Declaration

public ThirdPersonController otherPlayerPrefab
--

Field Value

TYPE	DESCRIPTION
ThirdPersonController	

otherPlayersContainer

Declaration

public GameObject otherPlayersContainer

Field Value

TYPE	DESCRIPTION
GameObject	

playerDataManager

Declaration

public PlayerDataManager playerDataManager
--

Field Value

TYPE	DESCRIPTION
PlayerDataManager	

playersInstantiated

Declaration

public Dictionary<string, bool> playersInstantiated

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Boolean>	

ratePetiton

Declaration

public float ratePetiton

Field Value

TYPE	DESCRIPTION
System.Single	

timer

Declaration

public float timer

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

CurrentPlayer

Declaration

```
public ThirdPersonController CurrentPlayer { get; set; }
```

Property Value

TYPE	DESCRIPTION
ThirdPersonController	

CurrentSecondsBetweenPetition

Declaration

```
public float CurrentSecondsBetweenPetition { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

OnConnected

Declaration

```
public Action<ThirdPersonController> OnConnected { get; set; }
```

Property Value

TYPE	DESCRIPTION
Action< ThirdPersonController >	

Methods

CheckIfIsMoving(LobbyPlayerBasePayload, LobbyPlayerBasePayload)

Declaration

```
public bool CheckIfIsMoving(LobbyPlayerBasePayload currentData, LobbyPlayerBasePayload dataToSend)
```

Parameters

TYPE	NAME	DESCRIPTION
LobbyPlayerBasePayload	currentData	
LobbyPlayerBasePayload	dataToSend	

Returns

TYPE	DESCRIPTION
System.Boolean	

CheckIfOtherPlayerHasChallengeForPlayfabId(String)

Declaration

```
public bool CheckIfOtherPlayerHasChallengeForPlayfabId(string playfabId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playfabId	

Returns

TYPE	DESCRIPTION
System.Boolean	

CheckIfPlayerHasChallengeOrIsChallenging(String)

Declaration

```
public bool CheckIfPlayerHasChallengeOrIsChallenging(string playfabId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playfabId	

Returns

TYPE	DESCRIPTION
System.Boolean	

ConnectToLobby(Int32)

Declaration

```
public void ConnectToLobby(int lobbyId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	lobbyId	

GeneratePlayfabIdHash(String)

Declaration

```
public string GeneratePlayfabIdHash(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	

Returns

TYPE	DESCRIPTION
System.String	

GetPlayerState(String)

Declaration

```
public CharacterStateEnum GetPlayerState(string playerId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playerId	

Returns

TYPE	DESCRIPTION
CharacterStateEnum	

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

ManageLobbyPlayers(Dictionary<String, LobbyPlayerBasePayload>)

Declaration

```
public void ManageLobbyPlayers(Dictionary<string, LobbyPlayerBasePayload> usersData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, LobbyPlayerBasePayload>	usersData	

NewPlayerConnected(KeyValuePair<String, LobbyPlayerBasePayload>)

Declaration

```
public void NewPlayerConnected(KeyValuePair<string, LobbyPlayerBasePayload> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<System.String, LobbyPlayerBasePayload>	item	

OnApplicationQuit()

Declaration

```
public void OnApplicationQuit()
```

PaintOtherPlayer(CharacterRecolor, Dictionary<PartsOfCharacter, CharacterStyleInfo>)

Declaration

```
public void PaintOtherPlayer(CharacterRecolor playerRecolor, Dictionary<PartsOfCharacter, CharacterStyleInfo> characterStyle)
```

Parameters

TYPE	NAME	DESCRIPTION
CharacterRecolor	playerRecolor	
System.Collections.Generic.Dictionary<PartsOfCharacter, CharacterStyleInfo >	characterStyle	

PlayerChallengeVerification(KeyValuePair<String, LobbyPlayerBasePayload>)

Declaration

```
public void PlayerChallengeVerification(KeyValuePair<string, LobbyPlayerBasePayload> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<System.String, LobbyPlayerBasePayload>	item	

ShowPlayerPersonalUI(Boolean)

Declaration

```
public void ShowPlayerPersonalUI(bool show = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	show	

Start()

Declaration

```
public void Start()
```

TickCheck()

Declaration

```
public void TickCheck()
```

TickCheckActivates()

Declaration

```
public void TickCheckActivates()
```

ToggleBattleMode(Boolean)

Declaration

```
public void ToggleBattleMode(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

ToogleChatMode()

Declaration

```
public void ToogleChatMode()
```

UntoogleChatMode()

Declaration

```
public void UntoogleChatMode()
```

Implements

[ILobbyNetworkManager](#)

IService

Class LobbyNetworkingController.DirectMessage

Inheritance

System.Object

LobbyNetworkingController.DirectMessage

Namespace: [Timba.Games.SacredTails.LobbyNetworking](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DirectMessage
```

Fields

from

Declaration

```
public string from
```

Field Value

TYPE	DESCRIPTION
System.String	

msg

Declaration

```
public string msg
```

Field Value

TYPE	DESCRIPTION
System.String	

Class LobbyNetworkingController.MessageObject

Inheritance

System.Object

LobbyNetworkingController.MessageObject

Namespace: [Timba.Games.SacredTails.LobbyNetworking](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class MessageObject
```

Fields

messages

Declaration

```
public List<LobbyNetworkingController.DirectMessage> messages
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< LobbyNetworkingController.DirectMessage >	

Class PlayersStreamData

Inheritance

System.Object

PlayersStreamData

Namespace: [Timba.Games.SacredTails.LobbyNetworking](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayersStreamData
```

Fields

LastUpdated

Declaration

```
public string LastUpdated
```

Field Value

TYPE	DESCRIPTION
System.String	

Permission

Declaration

```
public string Permission
```

Field Value

TYPE	DESCRIPTION
System.String	

Value

Declaration

```
public string Value
```

Field Value

TYPE	DESCRIPTION
System.String	

Namespace Timba.Games.SacredTails.PopupModule

Classes

[PopupManager](#)

Interfaces

[IPopupManager](#)

Enums

[PopupManager.ButtonType](#)

Interface IPopupManager

Namespace: [Timba.Games.SacredTails.PopupModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface IPopupManager
```

Methods

HideInfoPopup()

Declaration

```
void HideInfoPopup()
```

ShowInfoPopup(String, Dictionary<PopupManager.ButtonType, Action>)

Declaration

```
void ShowInfoPopup(string textInfo, Dictionary<PopupManager.ButtonType, Action> buttonsActionsPair = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textInfo	
System.Collections.Generic.Dictionary< PopupManager.ButtonType , Action>	buttonsActionsPair	

Class PopupManager

Inheritance

System.Object
PopupManager

Implements

[IPopupManager](#)
IService

Namespace: [Timba.Games.SacredTails.PopupModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PopupManager : MonoBehaviour, IPopupManager
```

Methods

HideInfoPopup()

Declaration

```
public void HideInfoPopup()
```

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

ShowInfoPopup(String, Dictionary<PopupManager.ButtonType, Action>)

Declaration

```
public void ShowInfoPopup(string textInfo, Dictionary<PopupManager.ButtonType, Action> buttonsActionsPair = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textInfo	
System.Collections.Generic.Dictionary< PopupManager.ButtonType , Action>	buttonsActionsPair	

Implements

[IPopupManager](#)
IService

Enum PopupManager.ButtonType

Namespace: [Timba.Games.SacredTails.PopupModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public enum ButtonType
```

Fields

NAME	DESCRIPTION
BACK_BUTTON	
CONFIRM_BUTTON	

Namespace Timba.Games.SacredTails.StoreModule

Classes

[StoreController](#)

[StoreItem](#)

[StoreView](#)

Structs

[StoreItemData](#)

Class StoreController

Inheritance

System.Object

StoreController

Namespace: [Timba.Games.SacredTails.StoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class StoreController : MonoBehaviour
```

Methods

BuyItem(Int32, UInt32)

Declaration

```
public void BuyItem(int itemId, uint itemPrice)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	itemId	
System.UInt32	itemPrice	

CreatePool()

Declaration

```
public void CreatePool()
```

FillPool(List<CatalogItem>)

Declaration

```
public void FillPool(List<CatalogItem> items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<CatalogItem>	items	

HideStore()

Declaration

```
public void HideStore()
```

RequestCardsStore()

Declaration

```
public void RequestCardsStore()
```

SetCardStore(GetCatalogItemsResult)

Declaration

```
public void SetCardStore(GetCatalogItemsResult result)
```

Parameters

TYPE	NAME	DESCRIPTION
GetCatalogItemsResult	result	

Start()

Declaration

```
public void Start()
```

SubscribeToPurchaseEvent()

Declaration

```
public void SubscribeToPurchaseEvent()
```

Class StoreItem

Inheritance

System.Object

StoreItem

Namespace: [Timba.Games.SacredTails.StoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class StoreItem : MonoBehaviour
```

Fields

cardButton

Declaration

```
public Button cardButton
```

Field Value

TYPE	DESCRIPTION
Button	

data

Declaration

```
public StoreItemData data
```

Field Value

TYPE	DESCRIPTION
StoreItemData	

Methods

Init()

Declaration

```
public void Init()
```

SetCardButtonAction(Int32, Action<Int32>)

Declaration

```
public void SetCardButtonAction(int index, Action<int> callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	
Action<System.Int32>	callback	

Struct StoreItemData

Namespace: [Timba.Games.SacredTails.StoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public struct StoreItemData
```

Fields

itemId

Declaration

```
public int itemId
```

Field Value

TYPE	DESCRIPTION
System.Int32	

itemName

Declaration

```
public string itemName
```

Field Value

TYPE	DESCRIPTION
System.String	

itemPrice

Declaration

```
public uint itemPrice
```

Field Value

TYPE	DESCRIPTION
System.UInt32	

itemSprite

Declaration

```
public Sprite itemSprite
```

Field Value

TYPE	DESCRIPTION
Sprite	

priceCurrency

Declaration

public string priceCurrency

Field Value

TYPE	DESCRIPTION
System.String	

Class StoreView

Inheritance

System.Object

StoreView

Namespace: [Timba.Games.SacredTails.StoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class StoreView : MonoBehaviour
```

Methods

HidePanelAfterPurchase()

Declaration

```
public void HidePanelAfterPurchase()
```

HidePopup()

Declaration

```
public void HidePopup()
```

HideStore(List<StoreItem>)

Declaration

```
public void HideStore(List<StoreItem> storeItems)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< StoreItem >	storeItems	

ShowPopupPurchaseCard(Int32, UInt32, Action<Int32, UInt32>)

Declaration

```
public void ShowPopupPurchaseCard(int itemId, uint itemPrice, Action<int, uint> onPurchaseCard)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	itemId	
System.UInt32	itemPrice	
Action<System.Int32, System.UInt32>	onPurchaseCard	

ShowPopupPurchaseResult(String)

Declaration

```
public void ShowPopupPurchaseResult(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

Namespace Timba.Games.SacredTails.WalletModule

Classes

[WalletView](#)

Class WalletView

Inheritance

System.Object

WalletView

Namespace: [Timba.Games.SacredTails.WalletModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WalletView : MonoBehaviour
```

Fields

currencyText

Declaration

```
public TMP_Text currencyText
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

walletPanel

Declaration

```
public GameObject walletPanel
```

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

ChangeCurrency(Int32)

Declaration

```
public void ChangeCurrency(int currentCurrency)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	currentCurrency	

HideUserWallet()

Declaration

```
public void HideUserWallet()
```

ShowUserWallet()

Declaration

```
public void ShowUserWallet()
```

Namespace Timba.Packages.Games.ChatModule.Controller

Classes

[LobbyChatController](#)

Class LobbyChatController

Inheritance

System.Object

LobbyChatController

Namespace: [Timba.Packages.Games.ChatModule.Controller](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LobbyChatController : LobbyChatCoreController
```

Methods

Init<T>(T)

Declaration

```
public override void Init<T>(T data)
```

Parameters

TYPE	NAME	DESCRIPTION
T	data	

Type Parameters

NAME	DESCRIPTION
T	

Namespace Timba.Packages.Games.ChatModule.Core

Classes

[LobbyChatCoreController](#)

Interfaces

[ILobbyChatViewable](#)

Interface ILobbyChatViewable

Namespace: [Timba.Packages.Games.ChatModule.Core](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface ILobbyChatViewable
```

Properties

OnSendMessage

Declaration

```
Action<string> OnSendMessage { get; set; }
```

Property Value

TYPE	DESCRIPTION
Action<System.String>	

Methods

Init()

Declaration

```
void Init()
```

ShowMessage(MessageDto)

Declaration

```
void ShowMessage(MessageDto MessageDto)
```

Parameters

TYPE	NAME	DESCRIPTION
MessageDto	MessageDto	

Class LobbyChatCoreController

Inheritance

System.Object

LobbyChatCoreController

Namespace: [Timba.Packages.Games.ChatModule.Core](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class LobbyChatCoreController : MonoBehaviour
```

Fields

chatView

Declaration

```
protected ILobbyChatViewable chatView
```

Field Value

TYPE	DESCRIPTION
ILobbyChatViewable	

Methods

Init<T>(T)

Declaration

```
public abstract void Init<T>(T data)
```

Parameters

TYPE	NAME	DESCRIPTION
T	data	

Type Parameters

NAME	DESCRIPTION
T	

Namespace Timba.Packages.Games.ChatModule.Model

Classes

[LobbyChatModel](#)

Class LobbyChatModel

Inheritance

System.Object

LobbyChatModel

Namespace: [Timba.Packages.Games.ChatModule.Model](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class LobbyChatModel
```

Fields

OnChatMessage

Declaration

```
public static Action<MessageDto> OnChatMessage
```

Field Value

TYPE	DESCRIPTION
Action<MessageDto>	

Methods

CMDSendMessage(MessagePayload, Action<Boolean>)

Declaration

```
public static void CMDSendMessage(MessagePayload message, Action<bool> callback)
```

Parameters

TYPE	NAME	DESCRIPTION
MessagePayload	message	
Action<System.Boolean>	callback	

TestMessage(String, String)

Declaration

```
public static void TestMessage(string _message, string _nickname)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_message	
System.String	_nickname	

Namespace Timba.Packages.Games.ChatModule.Structs

Structs

[MessageDto](#)

[MessagePayload](#)

Struct MessageDto

Namespace: [Timba.Packages.Games.ChatModule.Structs](#)

Assembly: cs.temp.dll.dll

Syntax

```
public struct MessageDto
```

Fields

message

Declaration

```
public string message
```

Field Value

TYPE	DESCRIPTION
System.String	

nickname

Declaration

```
public string nickname
```

Field Value

TYPE	DESCRIPTION
System.String	

Struct MessagePayload

Namespace: [Timba.Packages.Games.ChatModule.Structs](#)

Assembly: cs.temp.dll.dll

Syntax

```
public struct MessagePayload
```

Fields

messageString

Declaration

```
public string messageString
```

Field Value

TYPE	DESCRIPTION
System.String	

userId

Declaration

```
public string userId
```

Field Value

TYPE	DESCRIPTION
System.String	

Namespace Timba.Packages.Games.PlayerControllerModule

Classes

[InputHandler](#)

[InputSource](#)

Interfaces

[IInputHandleable](#)

Interface IInputHandleable

Namespace: [Timba.Packages.Games.PlayerControllerModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface IInputHandleable
```

Methods

GetHorizontalInput()

Declaration

```
float GetHorizontalInput()
```

Returns

TYPE	DESCRIPTION
System.Single	

GetVerticalInput()

Declaration

```
float GetVerticalInput()
```

Returns

TYPE	DESCRIPTION
System.Single	

Init()

Declaration

```
void Init()
```

Class InputHandler

Inheritance
System.Object
InputHandler

Implements
[IInputHandleable](#)

Namespace: [Timba.Packages.Games.PlayerControllerModule](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class InputHandler : MonoBehaviour, IInputHandleable
```

Fields

keyBindings

Declaration

```
public InputSource keyBindings
```

Field Value

TYPE	DESCRIPTION
InputSource	

Methods

GetHorizontalInput()

Declaration

```
public float GetHorizontalInput()
```

Returns

TYPE	DESCRIPTION
System.Single	

GetInteractableInput()

Declaration

```
public bool GetInteractableInput()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

GetVerticalInput()

Declaration

```
public float GetVerticalInput()
```

Returns

TYPE	DESCRIPTION
System.Single	

Init()

Declaration

```
public void Init()
```

Implements

[IInputHandleable](#)

Class InputSource

Inheritance

System.Object

InputSource

Namespace: [Timba.Packages.Games.PlayerControllerModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class InputSource : ScriptableObject
```

Fields

down

Declaration

```
public KeyCode[] down
```

Field Value

TYPE	DESCRIPTION
KeyCode[]	

interact

Declaration

```
public KeyCode[] interact
```

Field Value

TYPE	DESCRIPTION
KeyCode[]	

left

Declaration

```
public KeyCode[] left
```

Field Value

TYPE	DESCRIPTION
KeyCode[]	

right

Declaration

```
public KeyCode[] right
```

Field Value

TYPE	DESCRIPTION
KeyCode[]	

up

Declaration

```
public KeyCode[] up
```

Field Value

TYPE	DESCRIPTION
KeyCode[]	

Methods

SetInteractValue()

Declaration

```
public bool SetInteractValue()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

SetMoveValue(KeyCode[], KeyCode[])

Declaration

```
public float SetMoveValue(KeyCode[] aSet, KeyCode[] bSet)
```

Parameters

TYPE	NAME	DESCRIPTION
KeyCode[]	aSet	
KeyCode[]	bSet	

Returns

TYPE	DESCRIPTION
System.Single	

Namespace

Timba.Packages.Games.PlayerControllerModule.Core

Classes

[ThirdPersonCoreController](#)

Class ThirdPersonCoreController

Inheritance

System.Object

ThirdPersonCoreController

Namespace: [Timba.Packages.Games.PlayerControllerModule.Core](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class ThirdPersonCoreController : MonoBehaviour
```

Fields

inputHandler

Declaration

```
protected IInputHandleable inputHandler
```

Field Value

TYPE	DESCRIPTION
IInputHandleable	

Methods

Init<T>(T)

Declaration

```
public abstract void Init<T>(T data)
```

Parameters

TYPE	NAME	DESCRIPTION
T	data	

Type Parameters

NAME	DESCRIPTION
T	

Namespace Timba.Patterns.ServiceLocator

Classes

[ServiceLocator](#)

[ServiceLocatorConfig](#)

[SingleInstanceMonoBehaviour<T>](#)

[SingletonMonoBehaviour<T>](#)

Avoid using singletons. If you need a class that guarantees a single instance consider using [SingleInstanceMonoBehaviour](#) or [SingletonMonoBehaviour](#). If you must have a singleton, consider registering it as a service instead using [ServiceLocator](#)

Interfaces

[IService](#)

Interface IService

Namespace: [Timba.Patterns.ServiceLocator](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface IService
```

Properties

gameObject

Declaration

```
GameObject gameObject { get; }
```

Property Value

TYPE	DESCRIPTION
GameObject	

Methods

IsReady()

Declaration

```
bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Class ServiceLocator

Inheritance

System.Object
[SingletonMonoBehaviour<ServiceLocator>](#)
ServiceLocator

Inherited Members

[SingletonMonoBehaviour<ServiceLocator>.Instance](#)
[SingletonMonoBehaviour<ServiceLocator>.Awake\(\)](#)
[SingletonMonoBehaviour<ServiceLocator>.initialized](#)

Namespace: [Timba.Patterns.ServiceLocator](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class ServiceLocator : SingletonMonoBehaviour<ServiceLocator>
```

Fields

config

Declaration

```
public ServiceLocatorConfig config
```

Field Value

TYPE	DESCRIPTION
ServiceLocatorConfig	

Methods

GetService<T>()

Finds a service of type T. The search is done in the following order: 1 - Look for a registered service in the Service Locator 2 - Find an existing game object of type T in the scene 3 - Find a default service of type T in ServiceLocatorConfig This call can be slow. Always keep a local reference of the service you find

Declaration

```
public T GetService<T>()  
    where T : IService
```

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	Type of the service to find

Initialize()

Declaration

```
protected override void Initialize()
```

Overrides

```
Timba.Patterns.ServiceLocator.SingletonMonoBehaviour<Timba.Patterns.ServiceLocator.ServiceLocator>.Initialize()
```

Class ServiceLocatorConfig

Inheritance

System.Object
ServiceLocatorConfig

Namespace: [Timba.Patterns.ServiceLocator](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ServiceLocatorConfig : ScriptableObject
```

Fields

defaultServicesPrefabs

Declaration

```
public GameObject[] defaultServicesPrefabs
```

Field Value

TYPE	DESCRIPTION
GameObject[]	

Class SingleInstanceMonoBehaviour<T>

Inheritance

System.Object

SingleInstanceMonoBehaviour<T>

Namespace: [Timba.Patterns.ServiceLocator](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SingleInstanceMonoBehaviour<T> : MonoBehaviour where T : SingleInstanceMonoBehaviour<T>
```

Type Parameters

NAME	DESCRIPTION
T	

Fields

initialized

Declaration

```
protected bool initialized
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Awake()

Declaration

```
public virtual void Awake()
```

Initialize()

Declaration

```
protected virtual void Initialize()
```

Class SingletonMonoBehaviour<T>

Avoid using singletons. If you need a class that guarantees a single instance consider using `SingleInstanceMonoBehaviour` or simmlar If you must have a singleton, consider registering it as a service instead using `ServiceLocator`

Inheritance

System.Object

SingletonMonoBehaviour<T>

[ServiceLocator](#)

Namespace: [Timba.Patterns.ServiceLocator](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SingletonMonoBehaviour<T> : MonoBehaviour where T : SingletonMonoBehaviour<T>
```

Type Parameters

NAME	DESCRIPTION
T	

Fields

initialized

Declaration

```
protected bool initialized
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

Instance

Declaration

```
public static T Instance { get; }
```

Property Value

TYPE	DESCRIPTION
T	

Methods

Awake()

Declaration

```
public virtual void Awake()
```

Initialize()

Declaration

```
protected virtual void Initialize()
```


Namespace Timba.Recolor

Classes

[RecolorablePart3D](#)

Class RecolorablePart3D

Inheritance

System.Object

RecolorablePart3D

Namespace: [Timba.Recolor](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class RecolorablePart3D : MonoBehaviour
```

Fields

colors

Declaration

```
public Color32[] colors
```

Field Value

TYPE	DESCRIPTION
Color32[]	

Methods

SetColors(String[], Color32[])

Declaration

```
public void SetColors(string[] materialPropertyName, Color32[] newColors)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	materialPropertyName	
Color32[]	newColors	

Namespace Timba.SacredTails.Arena

Classes

[ActionCard](#)

Code representation of attack cards in the game

[ArenaShinseiSelectionController](#)

[ArenaShinseiSelectionUserPanelController](#)

Panel before the combat when players select the Shinsei party to the battle

[BattleGameMode](#)

[BattleShinseiSpawner](#)

Spawns Player's and Oponent's shinseis for the arena instances

[BattleStatisticsCalculator](#)

This class process all damage in the game apply the effect of statistics to final value, Attack, Deffence etc

[BattleStatisticsCalculatorSO](#)

Scriptable object for configurate the constants of the game for the BattleStatisticsCalculator

[BattleUIController](#)

Handles all of the events present in the UI of the Combat arena Mode

[BattleViewerController](#)

[CardDatabase](#)

[CardPreview](#)

UI element for card attack

[CardUI](#)

UI element of cards from the card store also show a preview of card with details

[Combat](#)

[CombatPlayer](#)

[CustomAtributes](#)

[GameSceneManager](#)

[ResourceBarValues](#)

[ShinseiPreviewPanelManager](#)

UI element that show stats and details of Shinsei

[ShinseiSlot](#)

[ShinseiStats](#)

[TextboxBase](#)

[Turn](#)

Turn describes the minimun necessary data to calculate battles in the game

TurnsController

This class handles the flow of the battle once it has started. Things like sending turns, receiving the match state and executing the turns with vfx are done/delegated here.

UserInfo

VfxInfo

Enums

AttacksAnimation

Class ActionCard

Code representation of attack cards in the game

Inheritance

System.Object
ActionCard

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class ActionCard : ScriptableObject
```

Fields

BattleAction

Declaration

```
public List<string> BattleAction
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

BattleActions

Declaration

```
public List<BattleActionData> BattleActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionData >	

cardImage

Declaration

```
public Sprite cardImage
```

Field Value

TYPE	DESCRIPTION
Sprite	

cardType

Declaration

```
public CharacterType cardType
```

Field Value

TYPE	DESCRIPTION
CharacterType	

casterAnimation

Declaration

public AttacksAnimation casterAnimation

Field Value

TYPE	DESCRIPTION
AttacksAnimation	

Description

Declaration

public string Description

Field Value

TYPE	DESCRIPTION
System.String	

DisplayNotification

Declaration

public string DisplayNotification

Field Value

TYPE	DESCRIPTION
System.String	

isComingFromCopyIndex

Declaration

public int isComingFromCopyIndex

Field Value

TYPE	DESCRIPTION
System.Int32	

PpCost

Declaration

public int PpCost

Field Value

TYPE	DESCRIPTION
System.Int32	

targetAnimation

Declaration

public AttacksAnimation targetAnimation

Field Value

TYPE	DESCRIPTION
AttacksAnimation	

vfxAffectBoth

Declaration

public bool vfxAffectBoth

Field Value

TYPE	DESCRIPTION
System.Boolean	

VfxIndex

Declaration

public int VfxIndex

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

OnValidate()

Declaration

public void OnValidate()

ShouldSerializeBattleActions()

Declaration

public bool ShouldSerializeBattleActions()
--

Returns

TYPE	DESCRIPTION
System.Boolean	

Class ArenaShinseiSelectionController

Inheritance

System.Object

ArenaShinseiSelectionController

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ArenaShinseiSelectionController : MonoBehaviour
```

Fields

OnShinseisSelected

Declaration

```
public Action<bool, List<int>> OnShinseisSelected
```

Field Value

TYPE	DESCRIPTION
Action<System.Boolean, System.Collections.Generic.List<System.Int32>>	

Methods

Init(MatchData, Int32, Boolean)

Declaration

```
public void Init(MatchData matchData, int playerIndex, bool isViewer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
MatchData	matchData	
System.Int32	playerIndex	
System.Boolean	isViewer	

OnShinseiSelectionCompleted()

Declaration

```
public void OnShinseiSelectionCompleted()
```


Class ArenaShinseiSelectionUserPanelController

Panel before the combat when players select the Shinsei party to the battle

Inheritance

System.Object

ArenaShinseiSelectionUserPanelController

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ArenaShinseiSelectionUserPanelController : MonoBehaviour
```

Fields

isEnemyPanel

Declaration

```
public bool isEnemyPanel
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

nameLabel

Declaration

```
public TMP_Text nameLabel
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

partyRow1

Declaration

```
public Transform partyRow1
```

Field Value

TYPE	DESCRIPTION
Transform	

partyRow2

Declaration

```
public Transform partyRow2
```

Field Value

TYPE	DESCRIPTION
Transform	

previewPanel

Declaration

```
public ShinseiPreviewPanelManager previewPanel
```

Field Value

TYPE	DESCRIPTION
ShinseiPreviewPanelManager	

selectedIndicator

Declaration

```
public Sprite selectedIndicator
```

Field Value

TYPE	DESCRIPTION
Sprite	

selectionIndicators

Declaration

```
public List<Image> selectionIndicators
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

selectionIndicatorsDetail

Declaration

```
public List<Image> selectionIndicatorsDetail
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

shinseSlotPrefab

Declaration

```
public ShinseiSlot shinseSlotPrefab
```

Field Value

TYPE	DESCRIPTION
ShinseiSlot	

textCounter

Declaration

public TextMeshProUGUI textCounter

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

unselectedIndicator

Declaration

public Sprite unselectedIndicator

Field Value

TYPE	DESCRIPTION
Sprite	

Methods

GetShinseisSelected()

Declaration

public List<int> GetShinseisSelected()
--

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

Init(CombatPlayer)

Declaration

public void Init(CombatPlayer combatPlayer)

Parameters

TYPE	NAME	DESCRIPTION
CombatPlayer	combatPlayer	

MakeSlotsUnclickeable()

Declaration

public void MakeSlotsUnclickeable()

SelectShinseiForBattle(Int32, ShinseiSlot)

Declaration

```
public void SelectShinseiForBattle(int listIndex, ShinseiSlot shinseiSlot)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	listIndex	
ShinseiSlot	shinseiSlot	

Show(Int32)

Declaration

```
public void Show(int direction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	direction	

ShowNext(Boolean)

Declaration

```
public void ShowNext(bool direction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	direction	

Enum AttacksAnimation

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public enum AttacksAnimation
```

Fields

NAME	DESCRIPTION
ATTACK0	
ATTACK1	
ATTACK2	
ATTACK3	
Awake	
Change	
Confused	
Damage	
Dance	
Death	
Dodge	
Emote	
NONE	
Ready	
Recharge	
Sleep	

Class BattleGameMode

Inheritance

System.Object

BattleGameMode

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleGameMode : MonoBehaviour
```

Fields

blackSprites

Declaration

```
public List<Sprite> blackSprites
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

cameraEndBattle

Declaration

```
public Camera cameraEndBattle
```

Field Value

TYPE	DESCRIPTION
Camera	

changeColorElements

Declaration

```
public List<Image> changeColorElements
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

changeColorInvertElements

Declaration

```
public List<Image> changeColorInvertElements
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

changeSpriteElements

Declaration

```
public List<Image> changeSpriteElements
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

colors

Declaration

```
public List<Color> colors
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Color>	

currentTerrain

Declaration

```
public BattleTerrainBehavioursBase currentTerrain
```

Field Value

TYPE	DESCRIPTION
BattleTerrainBehavioursBase	

delayTime

Declaration

```
public float delayTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

enemyInfo

Declaration

```
public UserInfo enemyInfo
```

Field Value

TYPE	DESCRIPTION
UserInfo	

gameSceneManager

Declaration

```
public GameManager gameSceneManager
```

Field Value

TYPE	DESCRIPTION
GameManager	

hasShowEndResult

Declaration

```
public bool hasShowEndResult
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isViewingMatch

Declaration

```
public bool isViewingMatch
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

localCombat

Declaration

```
public Combat localCombat
```

Field Value

TYPE	DESCRIPTION
Combat	

matchIdSeed

Declaration

```
public int matchIdSeed
```

Field Value

TYPE	DESCRIPTION
System.Int32	

playerInfo

Declaration


```
public UserInfo playerInfo
```

Field Value

TYPE	DESCRIPTION
UserInfo	

texts

Declaration

```
public List<TextMeshProUGUI> texts
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TextMeshProUGUI>	

titleEndMatchPanel

Declaration

```
public TMP_Text titleEndMatchPanel
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

turnsController

Declaration

```
public TurnsController turnsController
```

Field Value

TYPE	DESCRIPTION
TurnsController	

whiteSprites

Declaration

```
public List<Sprite> whiteSprites
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

Methods

AddTextToLog(String, Dictionary<String, String>)

Declaration

```
public void AddTextToLog(string text, Dictionary<string, string> customCodes = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	
System.Collections.Generic.Dictionary<System.String, System.String>	customCodes	

CancelSelectShinsei()

Declaration

```
public void CancelSelectShinsei()
```

EndMatchCheck(List<ResourceBarValues>)

Declaration

```
public bool EndMatchCheck(List<ResourceBarValues> healthBars)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< ResourceBarValues >	healthBars	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetCurrentShinseiEnergy()

Declaration

```
public int GetCurrentShinseiEnergy()
```

Returns

TYPE	DESCRIPTION
System.Int32	

GetOwnCombatData(Action<Combat>)

Declaration

```
public void GetOwnCombatData(Action<Combat> onRecieveData)
```

Parameters

TYPE	NAME	DESCRIPTION
Action< Combat >	onRecieveData	

GoBackToLobby()

Declaration

```
public void GoBackToLobby()
```

GoBackToLobbyPopup()

Declaration

```
public void GoBackToLobbyPopup()
```

InitBattle()

Declaration

```
public void InitBattle()
```

Initialize(MatchData)

Declaration

```
public void Initialize(MatchData matchData)
```

Parameters

TYPE	NAME	DESCRIPTION
MatchData	matchData	

MakeShinseiPartyUsingPlayfabId(String)

Bring shinsei party from playfab

Declaration

```
public void MakeShinseiPartyUsingPlayfabId(string playfabId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	playfabId	Playfab ID

OnApplicationQuit()

Declaration

```
public void OnApplicationQuit()
```

OnDisable()

Declaration

```
public void OnDisable()
```

OnStartMatch(GetMatchResult)

Declaration

```
public void OnStartMatch(GetMatchResult getMatchResult)
```

Parameters

TYPE	NAME	DESCRIPTION
GetMatchResult	getMatchResult	

OnStartMatchRoutine(GetMatchResult)

Declaration

```
public IEnumerator OnStartMatchRoutine(GetMatchResult getMatchResult)
```

Parameters

TYPE	NAME	DESCRIPTION
GetMatchResult	getMatchResult	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

OpenChangeShinseiPanel()

Declaration

```
public void OpenChangeShinseiPanel()
```

ShinseiSelected(Boolean, List<Int32>)

Declaration

```
public void ShinseiSelected(bool endTime, List<int> shinseisPlayer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	endTime	
System.Collections.Generic.List<System.Int32>	shinseisPlayer	

ShowEndResultPanel(Boolean, Action)

Declaration

```
public void ShowEndResultPanel(bool localPlayerWon, Action onEndAnimations = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	localPlayerWon	
Action	onEndAnimations	

ShowPopupEndMatch(Action, String, Boolean)

Declaration

```
public void ShowPopupEndMatch(Action callback, string message, bool localPlayerWon)
```

Parameters

TYPE	NAME	DESCRIPTION
Action	callback	
System.String	message	
System.Boolean	localPlayerWon	

SpawnWinnerShisneis(Boolean, Action)

Declaration

```
public void SpawnWinnerShisneis(bool localPlayerWon, Action onEndAnimations = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	localPlayerWon	
Action	onEndAnimations	

UpdateCurrentShinsei(ref UserInfo)

Declaration

```
public void UpdateCurrentShinsei(ref UserInfo userInfo)
```

Parameters

TYPE	NAME	DESCRIPTION
UserInfo	userInfo	

WaitForOpponentToSelect()

Declaration

```
public IEnumerator WaitForOpponentToSelect()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

WaitForOpponentToSelectViewMatch()

Declaration

```
public void WaitForOpponentToSelectViewMatch()
```

Class BattleShinseiSpawner

Spawns Player's and Oponent's shinseis for the arena instances

Inheritance

System.Object

BattleShinseiSpawner

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleShinseiSpawner : MonoBehaviour
```

Fields

enemyParent

Declaration

```
public Transform enemyParent
```

Field Value

TYPE	DESCRIPTION
Transform	

Methods

SpawnPlayerShinseis(Boolean, String)

Declaration

```
public CharacterSlot SpawnPlayerShinseis(bool isEnemy, string dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isEnemy	
System.String	dna	

Returns

TYPE	DESCRIPTION
CharacterSlot	

SpawnShinseiEndGame(List<String>, Transform)

Declaration

```
public void SpawnShinseiEndGame(List<string> dnas, Transform parent)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	dnas	
Transform	parent	

Class BattleStatisticsCalculator

This class process all damage in the game apply the effect of statistics to final value, Attack, Deffence etc

Inheritance

System.Object

BattleStatisticsCalculator

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public static class BattleStatisticsCalculator
```

Fields

healthMultiplier

Declaration

```
public static float healthMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_attackDivisionValue

Declaration

```
public static float m_attackDivisionValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_attackMultiplicationValue

Declaration

```
public static float m_attackMultiplicationValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_deffenseDivisionValue

Declaration

```
public static float m_deffenseDivisionValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_deffenseSumValue

Declaration

```
public static float m_deffenseSumValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_elementTypeBonusDivisionValue

Declaration

```
public static float m_elementTypeBonusDivisionValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_evasionThreshold1

Declaration

```
public static float m_evasionThreshold1
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_evasionThreshold2

Declaration

```
public static float m_evasionThreshold2
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_evasionThreshold3

Declaration

```
public static float m_evasionThreshold3
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_evasionThreshold4

Declaration

```
public static float m_evasionThreshold4
```

Field Value

TYPE	DESCRIPTION
System.Single	

m_evasionThreshold5

Declaration

```
public static float m_evasionThreshold5
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

ApplyEvationCritics(Int32)

Declaration

```
public static float ApplyEvationCritics(int multiplierStat)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	multiplierStat	

Returns

TYPE	DESCRIPTION
System.Single	

CalculateHealth(Int32)

Declaration

```
public static float CalculateHealth(int healthStat)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	healthStat	

Returns

TYPE	DESCRIPTION
System.Single	

CheckIfEvade(Int32, Single)

Declaration

```
public static bool CheckIfEvade(int multiplierStat, float evadeRoll)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	multiplierStat	
System.Single	evadeRoll	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetBonusStat(Shinsei, ShinseiStatsEnum, Single)

Declaration

```
public static float GetBonusStat(Shinsei shinsei, ShinseiStatsEnum statBonus, float bonusPercentage)
```

Parameters

TYPE	NAME	DESCRIPTION
Shinsei	shinsei	
ShinseiStatsEnum	statBonus	
System.Single	bonusPercentage	

Returns

TYPE	DESCRIPTION
System.Single	

GetDamageReceiveByTarget(Single, Single)

Declaration

```
public static float GetDamageReceiveByTarget(float deffenseStat, float rawDamage)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	deffenseStat	

TYPE	NAME	DESCRIPTION
System.Single	rawDamage	

Returns

TYPE	DESCRIPTION
System.Single	

GetRawDamage(Single, Single, Single, Single, Single)

Declaration

```
public static float GetRawDamage(float skillHitDamage, float shinseiAttackStat, float stab, float elementBonusMultiplier, float criticsMultiplier)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	skillHitDamage	
System.Single	shinseiAttackStat	
System.Single	stab	
System.Single	elementBonusMultiplier	
System.Single	criticsMultiplier	

Returns

TYPE	DESCRIPTION
System.Single	

GetStatByName(ShinseiStats, ShinseiStatsEnum)

Declaration

```
public static float GetStatByName(ShinseiStats shinseiStats, ShinseiStatsEnum statToModify)
```

Parameters

TYPE	NAME	DESCRIPTION
ShinseiStats	shinseiStats	
ShinseiStatsEnum	statToModify	

Returns

TYPE	DESCRIPTION
System.Single	

SetStatByName(ref ShinseiStats, ShinseiStatsEnum, Single, Boolean, Boolean, Boolean)

Declaration

```
public static (string, bool) SetStatByName(ref ShinseiStats shinseiStats, ShinseiStatsEnum statToModify, float amountToAdd, bool isPercentage, bool replaceStat = false, bool isEndAction = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ShinseiStats	shinseiStats	
ShinseiStatsEnum	statToModify	
System.Single	amountToAdd	
System.Boolean	isPercentage	
System.Boolean	replaceStat	
System.Boolean	isEndAction	

Returns

TYPE	DESCRIPTION
System.ValueTuple<System.String, System.Boolean>	

Class BattleStatisticsCalculatorSO

Scriptable object for configurate the constants of the game for the BattleStatisticsCalculator

Inheritance

System.Object

BattleStatisticsCalculatorSO

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class BattleStatisticsCalculatorSO : ScriptableObject
```

Fields

attackDivisionValue

Declaration

```
public float attackDivisionValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

attackMultiplicationValue

Declaration

```
public float attackMultiplicationValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

deffenseDivisionValue

Declaration

```
public float deffenseDivisionValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

deffenseSumValue

Declaration

```
public float deffenseSumValue
```

Field Value

TYPE	DESCRIPTION
System.Single	

elementTypeBonusDivison

Declaration

public float elementTypeBonusDivison

Field Value

TYPE	DESCRIPTION
System.Single	

evasionThreshold1

Declaration

public float evasionThreshold1

Field Value

TYPE	DESCRIPTION
System.Single	

evasionThreshold2

Declaration

public float evasionThreshold2

Field Value

TYPE	DESCRIPTION
System.Single	

evasionThreshold3

Declaration

public float evasionThreshold3

Field Value

TYPE	DESCRIPTION
System.Single	

evasionThreshold4

Declaration

public float evasionThreshold4

Field Value

TYPE	DESCRIPTION
System.Single	

evasionThreshold5

Declaration

```
public float evasionThreshold5
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

GetJsonActionCards()

Declaration

```
public void GetJsonActionCards()
```

SaveChanges()

Declaration

```
public void SaveChanges()
```


Class BattleUIController

Handles all of the events present in the UI of the Combat arena Mode

Inheritance

System.Object

BattleUIController

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleUIController : MonoBehaviour
```

Fields

battleMenu

Declaration

```
public GameObject battleMenu
```

Field Value

TYPE	DESCRIPTION
GameObject	

battleNotificationSystem

Declaration

```
public BattleNotificationSystem battleNotificationSystem
```

Field Value

TYPE	DESCRIPTION
BattleNotificationSystem	

cardContainer

Declaration

```
public GameObject cardContainer
```

Field Value

TYPE	DESCRIPTION
GameObject	

cardUis

Declaration

```
public List<CardUI> cardUis
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< CardUI >	

OnGetValueOfBars

Declaration

public Func<List<int>> OnGetValueOfBars

Field Value

TYPE	DESCRIPTION
Func<System.Collections.Generic.List<System.Int32>>	

titleEndMatchPanel

Declaration

public TMP_Text titleEndMatchPanel

Field Value

TYPE	DESCRIPTION
TMP_Text	

uIDisolver

Declaration

public UIDisolver uIDisolver

Field Value

TYPE	DESCRIPTION
UIDisolver	

versusPanelController

Declaration

public VersusPanelController versusPanelController
--

Field Value

TYPE	DESCRIPTION
VersusPanelController	

viewingBackToLobbyButton

Declaration

public GameObject viewingBackToLobbyButton
--

Field Value

TYPE	DESCRIPTION
GameObject	

Methods

ApplyEnergyChange(Int32, Int32)

Declaration

```
public void ApplyEnergyChange(int energyBarIndex, int newEnergyValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	energyBarIndex	
System.Int32	newEnergyValue	

ChangeHealthbarView()

Declaration

```
public void ChangeHealthbarView()
```

ChangeHealthColorBar(Int32)

Declaration

```
public void ChangeHealthColorBar(int indexBar)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexBar	

HideEverythingForWatchMatch(Boolean)

Declaration

```
public void HideEverythingForWatchMatch(bool isViewer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isViewer	

Init(ResourceBarValues, ResourceBarValues, ResourceBarValues, ResourceBarValues, String, String, Boolean)

Declaration

```
public void Init(ResourceBarValues player1Hp, ResourceBarValues player2Hp, ResourceBarValues player1Pp, ResourceBarValues player2Pp, string playerName, string enemyName, bool isViewing = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ResourceBarValues	player1Hp	
ResourceBarValues	player2Hp	
ResourceBarValues	player1Pp	
ResourceBarValues	player2Pp	
System.String	playerName	
System.String	enemyName	
System.Boolean	isViewing	

InitializeBars(Int32, Int32, Int32, Int32, Int32, Int32)

Declaration

```
public void InitializeBars(int playerHealth, int healthBarIndex, int maxValueHealth, int playerEnergy, int energyBarIndex, int maxValueEnergy)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	playerHealth	
System.Int32	healthBarIndex	
System.Int32	maxValueHealth	
System.Int32	playerEnergy	
System.Int32	energyBarIndex	
System.Int32	maxValueEnergy	

ShowCards(Boolean)

Declaration

```
public void ShowCards(bool isShow)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isShow	

ShowFaster(Int32)

Declaration

```
public void ShowFaster(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

ToggleWaitingPrompt(Boolean)

Declaration

```
public void ToggleWaitingPrompt(bool isWaiting)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isWaiting	

UpdateShinseiPicture(Int32, Shinsei)

Declaration

```
public void UpdateShinseiPicture(int playerIndex, Shinsei playerShinsei)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	playerIndex	
Shinsei	playerShinsei	

UpdateTimer(Single, String)

Declaration

```
public void UpdateTimer(float timeInSeconds, string colorText = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	timeInSeconds	
System.String	colorText	

Class BattleViewerController

Inheritance

System.Object

BattleViewerController

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BattleViewerController : MonoBehaviour
```

Fields

battleGameMode

Declaration

```
public BattleGameMode battleGameMode
```

Field Value

TYPE	DESCRIPTION
BattleGameMode	

Methods

Initialize(String, Boolean)

Declaration

```
public void Initialize(string matchId, bool isViewing)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
System.Boolean	isViewing	

Class CardDatabase

Inheritance

System.Object
CardDatabase

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CardDatabase : ScriptableObject
```

Fields

–

Declaration

```
public bool _
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

actionCards

Declaration

```
public List<ActionCard> actionCards
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ActionCard >	

searchByIndex

Declaration

```
public bool searchByIndex
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

searchByName

Declaration

```
public bool searchByName
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

searchByType

Declaration

```
public bool searchByType
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

GetJsonActionCards()

Declaration

```
public void GetJsonActionCards()
```

OnValidate()

Declaration

```
public void OnValidate()
```

SetVfxIndex()

Declaration

```
public void SetVfxIndex()
```


Class CardPreview

UI element for card attack

Inheritance

System.Object
CardPreview

Implements

IPointerClickHandler
IPointerDownHandler
IPointerEnterHandler
IPointerExitHandler

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CardPreview : MonoBehaviour
```

Fields

cardContainer

Declaration

```
public GameObject cardContainer
```

Field Value

TYPE	DESCRIPTION
GameObject	

cardCount

Declaration

```
public int cardCount
```

Field Value

TYPE	DESCRIPTION
System.Int32	

cardCountText

Declaration

```
public TMP_Text cardCountText
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

cardDesc

Declaration

```
public TMP_Text cardDesc
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

cardIndex

Declaration

```
public int cardIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

cardName

Declaration

```
public TMP_Text cardName
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

cardPP

Declaration

```
public TMP_Text cardPP
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

cardSprite

Declaration

```
public Image cardSprite
```

Field Value

TYPE	DESCRIPTION
Image	

cardType

Declaration

```
public Image cardType
```

Field Value

TYPE	DESCRIPTION
Image	

OnClickHandler

Declaration

<code>public Action<int> OnClickHandler</code>
--

Field Value

TYPE	DESCRIPTION
Action<System.Int32>	

onPointerDown

Declaration

<code>public Action onPointerDown</code>
--

Field Value

TYPE	DESCRIPTION
Action	

onPointerEnter

Declaration

<code>public Action onPointerEnter</code>

Field Value

TYPE	DESCRIPTION
Action	

onPointerExit

Declaration

<code>public Action onPointerExit</code>
--

Field Value

TYPE	DESCRIPTION
Action	

UseInCodeAkSounds

Declaration

<code>public bool UseInCodeAkSounds</code>
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

CopyCardIndex(CardPreview)

Declaration

```
public void CopyCardIndex(CardPreview cardPreview)
```

Parameters

TYPE	NAME	DESCRIPTION
CardPreview	cardPreview	

Init(ActionCard, Int32)

Declaration

```
public void Init(ActionCard card, int cardIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
ActionCard	card	
System.Int32	cardIndex	

OnPointerClick(PointerEventData)

Declaration

```
public void OnPointerClick(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

OnPointerDown(PointerEventData)

Declaration

```
public void OnPointerDown(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

OnPointerEnter(PointerEventData)

Declaration

```
public void OnPointerEnter(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

OnPointerExit(PointerEventData)

Declaration

```
public void OnPointerExit(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

SetToggable()

Declaration

```
public void SetToggable()
```

UpdateVisual(Int32)

Declaration

```
public void UpdateVisual(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Implements

IPointerClickHandler

IPointerDownHandler

IPointerEnterHandler

IPointerExitHandler

Class CardUI

UI element of cards from the card store also show a preview of card with details

Inheritance

System.Object

CardUI

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CardUI : MonoBehaviour
```

Fields

battleGameMode

Declaration

```
public BattleGameMode battleGameMode
```

Field Value

TYPE	DESCRIPTION
BattleGameMode	

textDescription

Declaration

```
public string textDescription
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

DisplayTooltip()

Declaration

```
public void DisplayTooltip()
```

OnTurnChange()

Declaration

```
public void OnTurnChange()
```

SetCardEmpty()

Declaration

```
public void SetCardEmpty()
```

SetDataCard(String, String, String, CharacterType)

Declaration

```
public void SetDataCard(string textTitle, string textDescription, string textEnergy, CharacterType cardType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textTitle	
System.String	textDescription	
System.String	textEnergy	
CharacterType	cardType	

Class Combat

Inheritance

System.Object

Combat

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class Combat
```

Fields

CurrentShinsei

Declaration

```
public int CurrentShinsei
```

Field Value

TYPE	DESCRIPTION
System.Int32	

CurrentTurn

Declaration

```
public int CurrentTurn
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MatchData

Declaration

```
public MatchData MatchData
```

Field Value

TYPE	DESCRIPTION
MatchData	

Turns

Declaration

```
public List<ActionCard> Turns
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ActionCard >	

Class CombatPlayer

Inheritance

System.Object

CombatPlayer

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class CombatPlayer
```

Fields

confirmState

Declaration

```
public bool confirmState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

DisplayName

Declaration

```
public string DisplayName
```

Field Value

TYPE	DESCRIPTION
System.String	

forbidenActions

Declaration

```
public Dictionary<int, int> forbidenActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, System.Int32>	

hasSurrender

Declaration

```
public bool hasSurrender
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

PlayfabId

Declaration

[NonSerialized] public string PlayfabId
--

Field Value

TYPE	DESCRIPTION
System.String	

ShinseiParty

Declaration

public List<Shinsei> ShinseiParty

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

shinseisSelected

Declaration

public bool shinseisSelected

Field Value

TYPE	DESCRIPTION
System.Boolean	

strikes

Declaration

public int strikes

Field Value

TYPE	DESCRIPTION
System.Int32	

Class CustomAtributes

Inheritance

System.Object

CustomAtributes

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class CustomAtributes
```

Fields

displayName

Declaration

```
public string displayName
```

Field Value

TYPE	DESCRIPTION
System.String	

PlayerPlayfabId

Declaration

```
public string PlayerPlayfabId
```

Field Value

TYPE	DESCRIPTION
System.String	

Skill

Declaration

```
public int Skill
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class GameManager

Inheritance

System.Object

GameManager

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class GameManager : NetworkBehaviour
```

Methods

EndBattle()

Declaration

```
public void EndBattle()
```

GetClosestBattleSpawn()

Declaration

```
public int GetClosestBattleSpawn()
```

Returns

TYPE	DESCRIPTION
System.Int32	

SendBattle(GetMatchResult, String, Int32, Boolean)

Declaration

```
public void SendBattle(GetMatchResult getMatchResult = null, string matchId = null, int _closestSpawnPoint = -1, bool isViewing = false)
```

Parameters

TYPE	NAME	DESCRIPTION
GetMatchResult	getMatchResult	
System.String	matchId	
System.Int32	_closestSpawnPoint	
System.Boolean	isViewing	

Class ResourceBarValues

Inheritance

System.Object

ResourceBarValues

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ResourceBarValues
```

Fields

currentValue

Declaration

```
public int currentValue
```

Field Value

TYPE	DESCRIPTION
System.Int32	

maxValue

Declaration

```
public int maxValue
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class ShinseiPreviewPanelManager

UI element that show stats and details of Shinsei

Inheritance

System.Object

ShinseiPreviewPanelManager

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiPreviewPanelManager : MonoBehaviour
```

Fields

backgroundTypeSwapper

Declaration

```
public BackgroundTypeSwapper backgroundTypeSwapper
```

Field Value

TYPE	DESCRIPTION
BackgroundTypeSwapper	

currentShinsei

Declaration

```
public Shinsei currentShinsei
```

Field Value

TYPE	DESCRIPTION
Shinsei	

database

Declaration

```
public IDatabase database
```

Field Value

TYPE	DESCRIPTION
IDatabase	

energy

Declaration

```
public Slider energy
```

Field Value

TYPE	DESCRIPTION
Slider	

energyLabel

Declaration

public TMP_Text energyLabel

Field Value

TYPE	DESCRIPTION
TMP_Text	

health

Declaration

public Slider health

Field Value

TYPE	DESCRIPTION
Slider	

healthLabel

Declaration

public TMP_Text healthLabel

Field Value

TYPE	DESCRIPTION
TMP_Text	

partElements

Declaration

public List<Image> partElements

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

positionCounter

Declaration

public PositionCounter positionCounter
--

Field Value

TYPE	DESCRIPTION
PositionCounter	

selectBtn

Declaration

public Button selectBtn

Field Value

TYPE	DESCRIPTION
Button	

shinseBackground

Declaration

public Image shinseBackground

Field Value

TYPE	DESCRIPTION
Image	

shinseiCards

Declaration

public List<CardPreview> shinseiCards

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< CardPreview >	

shinseiName

Declaration

public TMP_Text shinseiName

Field Value

TYPE	DESCRIPTION
TMP_Text	

shinseiSprite

Declaration

public Image shinseiSprite

Field Value

TYPE	DESCRIPTION
Image	

statLabels

Declaration

```
public List<TMP_Text> statLabels
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TMP_Text>	

uiHelper

Declaration

```
public IUIHelpable uiHelper
```

Field Value

TYPE	DESCRIPTION
IUIHelpable	

Methods

DisplayPreview(Shinsei, Boolean, Boolean, Boolean, Boolean, Int32)

Declaration

```
public void DisplayPreview(Shinsei shinsei, bool isEnemyPreview = false, bool isSelectionScreen = false, bool isVault = false, bool isCardManagement = false, int index = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
Shinsei	shinsei	
System.Boolean	isEnemyPreview	
System.Boolean	isSelectionScreen	
System.Boolean	isVault	
System.Boolean	isCardManagement	
System.Int32	index	

SetCardsPanel(Shinsei)

Declaration

```
public void SetCardsPanel(Shinsei shinsei)
```

Parameters

TYPE	NAME	DESCRIPTION
Shinsei	shinsei	

SetSlider(Slider, TMP_Text, Int32, Int32, Boolean)

Declaration

```
public void SetSlider(Slider slider, TMP_Text label, int value, int maxValue, bool isShinseiPreview = true)
```

Parameters

TYPE	NAME	DESCRIPTION
Slider	slider	
TMP_Text	label	
System.Int32	value	
System.Int32	maxValue	
System.Boolean	isShinseiPreview	

SetStatPanel(List<Single>, Dictionary<String, String>, CharacterType)

Declaration

```
public void SetStatPanel(List<float> shinseiStats, Dictionary<string, string> pTypes, CharacterType mainType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Single>	shinseiStats	
System.Collections.Generic.Dictionary<System.String, System.String>	pTypes	
CharacterType	mainType	

Class ShinseiSlot

Inheritance

System.Object
ShinseiSlot

Implements

IPointerClickHandler

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiSlot : MonoBehaviour
```

Fields

_energy

Declaration

```
public TMP_Text _energy
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

_energyBar

Declaration

```
public Slider _energyBar
```

Field Value

TYPE	DESCRIPTION
Slider	

_health

Declaration

```
public TMP_Text _health
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

_helathBar

Declaration

```
public Slider _helathBar
```

Field Value

TYPE	DESCRIPTION
Slider	

_highlightColor

Declaration

public Color _highlightColor

Field Value

TYPE	DESCRIPTION
Color	

_infoBtn

Declaration

public Button _infoBtn

Field Value

TYPE	DESCRIPTION
Button	

_shinseiName

Declaration

public TMP_Text _shinseiName

Field Value

TYPE	DESCRIPTION
TMP_Text	

_shinseiTypesImg

Declaration

public List<Image> _shinseiTypesImg

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Image>	

Counter

Declaration

public TextMeshProUGUI Counter

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

deactivateSlotOnClick

Declaration

public bool deactivateSlotOnClick

Field Value

TYPE	DESCRIPTION
System.Boolean	

isLocked

Declaration

public bool isLocked

Field Value

TYPE	DESCRIPTION
System.Boolean	

isPreviewOnly

Declaration

public bool isPreviewOnly

Field Value

TYPE	DESCRIPTION
System.Boolean	

listIndex

Declaration

public int listIndex

Field Value

TYPE	DESCRIPTION
System.Int32	

OnSlotClicked

Declaration

public UnityEvent<int, ShinseiSlot> OnSlotClicked

Field Value

TYPE	DESCRIPTION
UnityEvent<System.Int32, ShinseiSlot >	

previewBtn

Declaration

public Button previewBtn

Field Value

TYPE	DESCRIPTION
Button	

shinsei

Declaration

public Shinsei shinsei

Field Value

TYPE	DESCRIPTION
Shinsei	

shinseiKey

Declaration

public string shinseiKey

Field Value

TYPE	DESCRIPTION
System.String	

shinseiName

Declaration

public string shinseiName

Field Value

TYPE	DESCRIPTION
System.String	

shinseiView

Declaration

public Image shinseiView

Field Value

TYPE	DESCRIPTION
Image	

Properties

IsCompanion

Declaration

```
public bool IsCompanion { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

ChangeInteractable(Boolean, Boolean)

Declaration

```
public void ChangeInteractable(bool slotIsLocked = false, bool deactivateSlotOnClick = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	slotIsLocked	
System.Boolean	deactivateSlotOnClick	

ChangeShinseiSlotValues(String, Int32, Shinsei, String, ShinseiSlot)

Declaration

```
public void ChangeShinseiSlotValues(string shinseiKey = null, int listIndex = -1, Shinsei shinsei = null, string name = null, ShinseiSlot newSlot = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiKey	
System.Int32	listIndex	
Shinsei	shinsei	
System.String	name	
ShinseiSlot	newSlot	

OnPointerClick(PointerEventData)

Declaration

```
public void OnPointerClick(PointerEventData eventData)
```


Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

PopulateShinseiTypesSprites(String, CharacterType)

Declaration

```
public void PopulateShinseiTypesSprites(string dna, CharacterType mainType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	
CharacterType	mainType	

SetDataColor()

Declaration

```
public void SetDataColor()
```

UpdateVisual(String, String, Sprite)

Declaration

```
public void UpdateVisual(string shinseiName = null, string shinseiDNA = null, Sprite shinseiIcon = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiName	
System.String	shinseiDNA	
Sprite	shinseiIcon	

Implements

IPointerClickHandler

Class ShinseiStats

Inheritance

System.Object

ShinseiStats

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class ShinseiStats
```

Fields

Attack

Declaration

```
public float Attack
```

Field Value

TYPE	DESCRIPTION
System.Single	

Defence

Declaration

```
public float Defence
```

Field Value

TYPE	DESCRIPTION
System.Single	

Energy

Declaration

```
public int Energy
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Health

Declaration

```
public int Health
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Speed

Declaration

```
public float Speed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Stamina

Declaration

```
public float Stamina
```

Field Value

TYPE	DESCRIPTION
System.Single	

Vigor

Declaration

```
public float Vigor
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

attack

Declaration

```
public float attack { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

defence

Declaration

```
public float defence { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

speed

Declaration

public float speed { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

stamina

Declaration

public float stamina { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

vigor

Declaration

public float vigor { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

GetRealStat(Single)

Declaration

public float GetRealStat(float stat)

Parameters

TYPE	NAME	DESCRIPTION
System.Single	stat	

Returns

TYPE	DESCRIPTION
System.Single	

Class TextboxBase

Inheritance

System.Object

TextboxBase

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class TextboxBase : MonoBehaviour
```

Methods

AddRemoveFilter(String)

Declaration

```
public void AddRemoveFilter(string data)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	data	

AddText(String, Dictionary<String, String>)

Declaration

```
public virtual void AddText(string message, Dictionary<string, string> customCodes = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	
System.Collections.Generic.Dictionary<System.String, System.String>	customCodes	

ApplyFilters()

Declaration

```
public void ApplyFilters()
```

CheckAlteredState(String)

Declaration

```
public void CheckAlteredState(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

CheckIndividual(TextMeshProUGUI)

Declaration

```
public void CheckIndividual(TextMeshProUGUI text)
```

Parameters

TYPE	NAME	DESCRIPTION
TextMeshProUGUI	text	

State(Boolean)

Declaration

```
public void State(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

Class Turn

Turn describes the minimun necessary data to calculate battles in the game

Inheritance

System.Object

Turn

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class Turn
```

Fields

BattleActions

Declaration

```
public List<BattleActionData> BattleActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionData >	

indexCard

Declaration

```
public int indexCard
```

Field Value

TYPE	DESCRIPTION
System.Int32	

ppCost

Declaration

```
public int ppCost
```

Field Value

TYPE	DESCRIPTION
System.Int32	

turnActionType

Declaration

```
public CharacterType turnActionType
```

Field Value

TYPE	DESCRIPTION
CharacterType	

Class TurnsController

This class handles the flow of the battle once it has started. Things like sending turns, recieving the match state and executing the turns with vfx are done/delegated here.

Inheritance

System.Object
TurnsController

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TurnsController : MonoBehaviour
```

Fields

alteredStates

Declaration

```
public List<BattleAlteredStateBase> alteredStates
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleAlteredStateBase >	

battleEnemyCurrentActions

Declaration

```
public List<BattleActionData> battleEnemyCurrentActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionData >	

battlePlayerCurrentActions

Declaration

```
public List<BattleActionData> battlePlayerCurrentActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionData >	

calculateEndMatchAgain

Declaration

```
public bool calculateEndMatchAgain
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

camManager

Declaration

public CameraPlaneController camManager

Field Value

TYPE	DESCRIPTION
CameraPlaneController	

cardToTest

Declaration

public int cardToTest

Field Value

TYPE	DESCRIPTION
System.Int32	

deathTime

Declaration

public int deathTime

Field Value

TYPE	DESCRIPTION
System.Int32	

isEnemySleep

Declaration

public bool isEnemySleep

Field Value

TYPE	DESCRIPTION
System.Boolean	

isPlayerSleep

Declaration

public bool isPlayerSleep

Field Value

TYPE	DESCRIPTION
System.Boolean	

isViewingMatch

Declaration

public bool isViewingMatch

Field Value

TYPE	DESCRIPTION
System.Boolean	

localPlayer

Declaration

public CombatPlayer localPlayer

Field Value

TYPE	DESCRIPTION
CombatPlayer	

matchEnded

Declaration

public bool matchEnded

Field Value

TYPE	DESCRIPTION
System.Boolean	

testMatchSendTurnAutomatically

Declaration

public bool testMatchSendTurnAutomatically
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

turnActionsDatabase

Declaration

public List<BattleActionsBase> turnActionsDatabase
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionsBase >	

turnActionsDatabaseDictionary

Declaration

public Dictionary<ActionTypeEnum, BattleActionsBase> turnActionsDatabaseDictionary
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary< ActionTypeEnum , BattleActionsBase >	

uiCards

Declaration

public List<CardUI> uiCards

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< CardUI >	

vfxPositions

Declaration

public List<VFXPositionData> vfxPositions

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< VFXPositionData >	

vfxPositionsDictionary

Declaration

public Dictionary<VFXPositionEnum, Transform> vfxPositionsDictionary
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary< VFXPositionEnum , Transform>	

vfxsActionType

Declaration

public List<VFXTypeData> vfxsActionType

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< VFXTypeData >	

Methods

BtnSendTurn(Int32)

Method that will be called by the cards buttons. This method will start sending a turn to the server with the SendMyTurn method.

Declaration

```
public void BtnSendTurn(int indexCard)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexCard	

CalculateEnergyRegen(Shinsei, ResourceBarValues, List<BattleActionData>, Int32)

Calculates the amount of energy that a shinsei restores after the turn passes

Declaration

```
public void CalculateEnergyRegen(Shinsei playerShinsei, ResourceBarValues energyBar, List<BattleActionData> playersActions, int playerIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
Shinsei	playerShinsei	Shinsei of the player
ResourceBarValues	energyBar	Energy bar of the shinsei
System.Collections.Generic.List< BattleActionData >	playersActions	Player turn
System.Int32	playerIndex	Index of the player 0=local, 1=enemy

CalculateIncomingActions(ref List<BattleActionData>, Int32, Int32)

Execute the action of the target player on this turn, uses the battleaction dictionary to execute the pending actions from the player. Finally it checks if the shinseis are all dead in order to end the match.

Declaration

```
public void CalculateIncomingActions(ref List<BattleActionData> currentActionsData, int playerIndex, int ppCost)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< BattleActionData >	currentActionsData	Actions executed this turn
System.Int32	playerIndex	Index of user
System.Int32	ppCost	ppCost of card

CheckAlteredStates(Boolean, ActionCardDto)

Check the player altered states, execute them with the aid of the alteredStateControllers classes found in the alteredStates list.

Declaration

```
public bool CheckAlteredStates(bool isPlayer, ActionCardDto playerTurn)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isPlayer	Is the Local Player altered states
ActionCardDto	playerTurn	turn of the player checking the actions

Returns

TYPE	DESCRIPTION
System.Boolean	

CheckDeadTime(Int32, Int32)

Check the dead time in order to make the InitTurnsFlow() wait that time.

Declaration

```
public int CheckDeadTime(int healthOwner, int healthOther)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	healthOwner	health of current shinsei of the owner checking the actions
System.Int32	healthOther	helth of other player current shinsei

Returns

TYPE	DESCRIPTION
System.Int32	

CheckEndMatch(Boolean, UserInfo)

Check if the match has ended due to all the shinseis beign death

Declaration

public void CheckEndMatch(bool isLocalPlayer, UserInfo otherPlayer)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLocalPlayer	Is the player asking, the current player
UserInfo	otherPlayer	Player to check if shinseis are death

CheckEnergyBarsAndInitNextTurn()

After executing actions update energyBars and init next turn.

Declaration

public void CheckEnergyBarsAndInitNextTurn()
--

CheckIfAnyoneSurrendered(List<ActionCardDto>, Boolean)

Check if the enemy has surrender, if it has then end match.

Declaration

public bool CheckIfAnyoneSurrendered(List<ActionCardDto> playersTurnIndexCard, bool isLocalFirst)

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< ActionCardDto >	playersTurnIndexCard	Players turns
System.Boolean	isLocalFirst	Is the local player surrendering first

Returns

TYPE	DESCRIPTION
System.Boolean	

CurrentActionsHasSkipTurnForEnemy()

Checks the actions of both players to see if any of the both has a skip turn directed to the enemy player

Declaration

```
public bool CurrentActionsHasSkipTurnForEnemy()
```

Returns

TYPE	DESCRIPTION
System.Boolean	Enemy has skip turn

CurrentActionsHasSkipTurnForLocalPlayer()

Checks the actions of both players to see if any of the both has a skip turn directed to the local player

Declaration

```
public bool CurrentActionsHasSkipTurnForLocalPlayer()
```

Returns

TYPE	DESCRIPTION
System.Boolean	Player has skip turn

ExecuteServerAction(List<ActionCardDto>, Boolean)

Execute enemy actions that we brought earlier from the cloud.

Declaration

```
public Action ExecuteServerAction(List<ActionCardDto> turns, bool isEnemy = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< ActionCardDto >	turns	Turns of the players
System.Boolean	isEnemy	Is the enemy the one who want to execute his action?

Returns

TYPE	DESCRIPTION
Action	

ForbiddenActionCheck(ActionCard)

Declaration

```
public bool ForbiddenActionCheck(ActionCard actionCard)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
ActionCard	actionCard	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetTrueIndexCard(Int32)

Get the index of a card in the cardDatabase by passing the index of the card in the shinsei

Declaration

```
public int GetTrueIndexCard(int indexCard)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexCard	index of the card in the shinsei

Returns

TYPE	DESCRIPTION
System.Int32	

Examples

0,1,2,3 is for the first 4 cards of the shinsei. 4,5,6 is for the shinsei change 7 is for skipping turn 8 is for surrender

Init(BattleGameMode, Boolean)

Initialize turns controller by setting the vfx positions(enemy, center and player), then initialize a dictionary with of the battleActions(attack,buffdebuff,etc) Also it susbscribe to the change of turn to execute the InitNewTurn.

Declaration

```
public void Init(BattleGameMode battleGameMode, bool isViewing = false)
```

Parameters

TYPE	NAME	DESCRIPTION
BattleGameMode	battleGameMode	Reference to the battlemode in order to acces the users data i.e: shisneis
System.Boolean	isViewing	if the user is watching a match or participating in the match

InitCountDown()

Init turn countdown shown in the match to show max time before having a strike or ending the match by disconnection.

Declaration

```
public void InitCountDown()
```

InitNewTurn()

Init new turn checking if any player has a sleep in his battleActions in order to execute an automatic skipturn. This is temporal logic. The sleep logic should be it's own BattleAction/AlteredState.

Declaration

```
public void InitNewTurn()
```

InitTurnFlow(Boolean, Action, Action, Action)

Manage the turn execution flow executing the turns in the order the server executed them. Check if otherPlayer shinsei dies. If so, then doesn't execute his action.

Declaration

```
public void InitTurnFlow(bool ownerOfActionIsLocal, Action userAction, Action enemyAction, Action callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	ownerOfActionIsLocal	
Action	userAction	
Action	enemyAction	
Action	callback	

ReduceForbiddenActionsDuration()

Declaration

```
public void ReduceForbiddenActionsDuration()
```

SendMyTurn(Int32)

Prepare my turn with the card selected, execute it in local(cameras,ui) and send it to the other player. Also check if the card has been forbideen by other card effect, if so then shows a popup explaining that.

Declaration

```
public void SendMyTurn(int indexCard)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexCard	Index of card

SendTurnRequest(Int32, Int32)

Main function to send turn. Takes the index card and send a petition to the server to register this turn in the database. the server process it and then with the server response we continue to the WaitOponentTurn() method in order to wait for the other player to also send a turn

Declaration

```
public void SendTurnRequest(int indexCard, int timesTryingToSendTurn = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexCard	Index of the card to send in the shinsei card list i.e(0,1,2,3)
System.Int32	timesTryingToSendTurn	Times trying to send a turn, more than 5 is consider a player disconnect

SetForbiddenActions(ActionCard)

Declaration

```
public void SetForbiddenActions(ActionCard actionCard)
```

Parameters

TYPE	NAME	DESCRIPTION
ActionCard	actionCard	

TestSendTurn()

For testing only, this method will send an automatic turn to a bot player in the server.

Declaration

```
public void TestSendTurn()
```

WaitForSeconds(Single, Action)

Wait for X seconds before executing some logic

Declaration

```
public IEnumerator WaitForSeconds(float time, Action callback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	time	Time before executing logic
Action	callback	Logic to execute

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

WaitOponentTurn()

Bring data and actions from the server and execute them. Checks altered states, terrains and players actions and delegate all of those to be executed in order. It also checks if the match has already ended on the server in order to force the ending in the client too.

Declaration

```
public IEnumerator WaitOponentTurn()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Class UserInfo

Inheritance

System.Object

UserInfo

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UserInfo
```

Fields

battleShinseis

Declaration

```
public List<Shinsei> battleShinseis
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

currentShinseiIndex

Declaration

```
public int currentShinseiIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

energybars

Declaration

```
public List<ResourceBarValues> energybars
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ResourceBarValues >	

healthbars

Declaration

```
public List<ResourceBarValues> healthbars
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ResourceBarValues >	

isLocalPlayer

Declaration

```
public bool isLocalPlayer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

spawnedShinsei

Declaration

```
public CharacterSlot spawnedShinsei
```

Field Value

TYPE	DESCRIPTION
CharacterSlot	

turnActions

Declaration

```
public List<BattleActionsBase> turnActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< BattleActionsBase >	

userIndex

Declaration

```
public int userIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class VfxInfo

Inheritance

System.Object

VfxInfo

Namespace: [Timba.SacredTails.Arena](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VfxInfo : MonoBehaviour
```

Fields

isVfxReversed

Declaration

```
public bool isVfxReversed
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

vfxDuration

Declaration

```
public float vfxDuration
```

Field Value

TYPE	DESCRIPTION
System.Single	

Namespace Timba.SacredTails.Arena.ShinseiType

Classes

[ShinseiTypeListClass](#)

[ShinseiTypeScriptable](#)

Class ShinseiTypeListClass

Inheritance

System.Object
ShinseiTypeListClass

Namespace: [Timba.SacredTails.Arena.ShinseiType](#)
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class ShinseiTypeListClass
```

Fields

rows

Declaration

```
public List<float> rows
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Single>	

Class ShinseiTypeScriptable

Inheritance

System.Object
ShinseiTypeScriptable

Namespace: [Timba.SacredTails.Arena.ShinseiType](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class ShinseiTypeScriptable : ScriptableObject
```

Fields

shinseiTypeMatrix

Declaration

```
public List<ShinseiTypeListClass> shinseiTypeMatrix
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ShinseiTypeListClass >	

Methods

CompareTypesAndGetDamage(CharacterType, CharacterType)

Declaration

```
public float CompareTypesAndGetDamage(CharacterType currentShinsei, CharacterType targetShinesi)
```

Parameters

TYPE	NAME	DESCRIPTION
CharacterType	currentShinsei	
CharacterType	targetShinesi	

Returns

TYPE	DESCRIPTION
System.Single	

GetMessageForTypeDamage(Single)

Declaration

```
public string GetMessageForTypeDamage(float typeDamageMultiplier)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	typeDamageMultiplier	

Returns

TYPE	DESCRIPTION
System.String	

GetShinseiTypes()

Declaration

```
public void GetShinseiTypes()
```

SerializeShinseiTypeMatrix()

Declaration

```
public void SerializeShinseiTypeMatrix()
```

Namespace Timba.SacredTails.AudioIntegration

Classes

[AudioSettings](#)

[CallStart](#)

[SimpleAudioBtn](#)

This component add a callback to component attached button to play a Wwise event

[SimpleAudioCallback](#)

This component call an Wwise event

[SimpleAudioToggle](#)

UI Element that allow enable or disable audio in the game

[SimpleButtonSound](#)

Add callback to event OnClick of attached button using Unity Audio System

[SimpleCustomAudio](#)

Play simple custom audio

[WwiseSacred](#)

Class AudioSettings

Inheritance

System.Object

AudioSettings

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AudioSettings : MonoBehaviour
```

Fields

AmbienceVolume

Declaration

```
public AK.Wwise.RTPC AmbienceVolume
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.RTPC	

MasterSlider

Declaration

```
public Slider MasterSlider
```

Field Value

TYPE	DESCRIPTION
Slider	

MusicSlider

Declaration

```
public Slider MusicSlider
```

Field Value

TYPE	DESCRIPTION
Slider	

MusicVolume

Declaration

```
public AK.Wwise.RTPC MusicVolume
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.RTPC	

SFXSlider

Declaration

```
public Slider SFXSlider
```

Field Value

TYPE	DESCRIPTION
Slider	

SFXVolume

Declaration

```
public AK.Wwise.RTPC SFXVolume
```

Field Value

TYPE	DESCRIPTION
AK.Wwise.RTPC	

Methods

CustomStart()

Declaration

```
public void CustomStart()
```

SetMasterValue(Single)

Declaration

```
public void SetMasterValue(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	

SetMusicValue(Single)

Declaration

```
public void SetMusicValue(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	

SetSFXValue(Single)

Declaration

```
public void SetSFXValue(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	

Class CallStart

Inheritance

System.Object

CallStart

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CallStart : MonoBehaviour
```


Class SimpleAudioBtn

This component add a callback to component attached button to play a WWise event

Inheritance

System.Object

SimpleAudioBtn

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleAudioBtn : MonoBehaviour
```

Fields

eventName

Declaration

```
public string eventName
```

Field Value

TYPE	DESCRIPTION
System.String	

Class SimpleAudioCallback

This component call an WWise event

Inheritance

System.Object
SimpleAudioCallback

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleAudioCallback : MonoBehaviour
```

Fields

eventName

Declaration

```
public string eventName
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

PlayAudio()

Declaration

```
public void PlayAudio()
```

Class SimpleAudioToggle

UI Element that allow enable or disable audio in the game

Inheritance

System.Object

SimpleAudioToggle

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleAudioToggle : MonoBehaviour
```

Fields

eventName

Declaration

```
public string eventName
```

Field Value

TYPE	DESCRIPTION
System.String	

Class SimpleButtonSound

Add callback to event OnClick of attached button using Unity Audio System

Inheritance

System.Object

SimpleButtonSound

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleButtonSound : MonoBehaviour
```

Class SimpleCustomAudio

Play simple custom audio

Inheritance

System.Object

SimpleCustomAudio

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SimpleCustomAudio : MonoBehaviour
```

Fields

eventName

Declaration

```
public string eventName
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

Play()

Declaration

```
public void Play()
```

Class WwiseSacred

Inheritance

System.Object
WwiseSacred

Namespace: [Timba.SacredTails.AudioIntegration](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WwiseSacred : MonoBehaviour
```

Methods

PTEvent(String)

Declaration

```
public void PTEvent(string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	eventName	

Namespace Timba.SacredTails.BattleDebugTool

Classes

[ChangeCardsTest](#)

[DebugShinseiSlot](#)

Show all necesary information for debug battle comparing data from local, and server allow to find errors in code

[GetShareGroupData](#)

Download to client all data from the server for debugging

[GetShareGroupData.CombatPlayerServer](#)

Class ChangeCardsTest

Inheritance

System.Object

ChangeCardsTest

Namespace: [Timba.SacredTails.BattleDebugTool](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChangeCardsTest : MonoBehaviour
```

Fields

playerDataManager

Declaration

```
public PlayerDataManager playerDataManager
```

Field Value

TYPE	DESCRIPTION
PlayerDataManager	

Methods

ChangeTestCard0(String)

Declaration

```
public void ChangeTestCard0(string newIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newIndex	

ChangeTestCard1(String)

Declaration

```
public void ChangeTestCard1(string newIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newIndex	

ChangeTestCard2(String)

Declaration

```
public void ChangeTestCard2(string newIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newIndex	

ChangeTestCard3(String)

Declaration

```
public void ChangeTestCard3(string newIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newIndex	

Class DebugShinseiSlot

Show all necessary information for debug battle comparing data from local, and server allow to find errors in code

Inheritance

System.Object

DebugShinseiSlot

Namespace: [Timba.SacredTails.BattleDebugTool](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DebugShinseiSlot : MonoBehaviour
```

Fields

selectedImage

Declaration

```
public Image selectedImage
```

Field Value

TYPE	DESCRIPTION
Image	

Methods

ShowValues(ShinseiStats, Shinsei, Shinsei)

Declaration

```
public void ShowValues(ShinseiStats originalStats, Shinsei targetShinsei, Shinsei localShinsei)
```

Parameters

TYPE	NAME	DESCRIPTION
ShinseiStats	originalStats	
Shinsei	targetShinsei	
Shinsei	localShinsei	

Class GetShareGroupData

Download to client all data from the server for debugging

Inheritance

System.Object

GetShareGroupData

Namespace: [Timba.SacredTails.BattleDebugTool](#)

Assembly: [cs.temp.dll.dll](#)

Syntax

```
public class GetShareGroupData : MonoBehaviour
```

Fields

matchId

Declaration

```
public string matchId
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

GetServerMatchData(String, String)

Declaration

```
public void GetServerMatchData(string matchId, string ownPlayfabId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	matchId	
System.String	ownPlayfabId	

Class GetShareGroupData.CombatPlayerServer

Inheritance

System.Object

GetShareGroupData.CombatPlayerServer

Namespace: [Timba.SacredTails.BattleDebugTool](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class CombatPlayerServer
```

Fields

confirmState

Declaration

```
public bool confirmState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

currentShinsei

Declaration

```
public int currentShinsei
```

Field Value

TYPE	DESCRIPTION
System.Int32	

DisplayName

Declaration

```
public string DisplayName
```

Field Value

TYPE	DESCRIPTION
System.String	

forbidenActions

Declaration

```
public Dictionary<int, int> forbidenActions
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, System.Int32>	

hasSurrender

Declaration

public bool hasSurrender

Field Value

TYPE	DESCRIPTION
System.Boolean	

playfabId

Declaration

public string playfabId

Field Value

TYPE	DESCRIPTION
System.String	

ShinseiParty

Declaration

public List<Shinsei> ShinseiParty

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Shinsei >	

shinseisSelected

Declaration

public bool shinseisSelected

Field Value

TYPE	DESCRIPTION
System.Boolean	

strikes

Declaration

public int strikes

Field Value

TYPE	DESCRIPTION
System.Int32	

Namespace Timba.SacredTails.CameraModule

Classes

[CameraTriggerChange](#)

[ViewerCameraChange](#)

Class CameraTriggerChange

Inheritance

System.Object

CameraTriggerChange

Namespace: [Timba.SacredTails.CameraModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CameraTriggerChange : MonoBehaviour
```

Fields

lobbyCamera

Declaration

```
public CinemachineVirtualCamera lobbyCamera
```

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

treeCamera

Declaration

```
public CinemachineVirtualCamera treeCamera
```

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

Methods

OnTriggerEnter(Collider)

Declaration

```
public void OnTriggerEnter(Collider other)
```

Parameters

TYPE	NAME	DESCRIPTION
Collider	other	

OnTriggerExit(Collider)

Declaration

```
public void OnTriggerExit(Collider other)
```

Parameters

TYPE	NAME	DESCRIPTION
Collider	other	

Class ViewerCameraChange

Inheritance

System.Object

ViewerCameraChange

Namespace: [Timba.SacredTails.CameraModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ViewerCameraChange : MonoBehaviour
```

Fields

lobbyCamera

Declaration

```
public CinemachineVirtualCamera lobbyCamera
```

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

viewerCamera

Declaration

```
public CinemachineVirtualCamera viewerCamera
```

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

Methods

ChangeCamera()

Declaration

```
public void ChangeCamera()
```

Namespace Timba.SacredTails.CardStoreModule

Classes

[Card](#)

[CardManagementController](#)

[CardView](#)

[Deck](#)

Class Card

Inheritance

System.Object
Card

Namespace: [Timba.SacredTails.CardStoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class Card
```

Fields

cardName

Declaration

```
public string cardName
```

Field Value

TYPE	DESCRIPTION
System.String	

count

Declaration

```
public int count
```

Field Value

TYPE	DESCRIPTION
System.Int32	

index

Declaration

```
public int index
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class CardManagementController

Inheritance

System.Object

CardManagementController

Namespace: [Timba.SacredTails.CardStoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CardManagementController : MonoBehaviour
```

Fields

backgroundTypeSwapper

Declaration

```
public BackgroundTypeSwapper backgroundTypeSwapper
```

Field Value

TYPE	DESCRIPTION
BackgroundTypeSwapper	

Methods

AddCardInDeck(Int32)

Declaration

```
public void AddCardInDeck(int indexCardToAdd)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexCardToAdd	

AddSelectedCard(Int32)

Declaration

```
public void AddSelectedCard(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

ChangeTargetShinseiCard(Int32)

Declaration

```
public void ChangeTargetShinseiCard(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

DeselectTargetCard()

Declaration

```
public void DeselectTargetCard()
```

InitCardManager()

Declaration

```
public void InitCardManager()
```

RemoveCardInDeck(Int32)

Declaration

```
public void RemoveCardInDeck(int indexCardToAdd)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	indexCardToAdd	

RemoveSelectedCard(Int32)

Declaration

```
public void RemoveSelectedCard(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

SelectTargetCard(Int32)

Declaration

```
public void SelectTargetCard(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

SetCurrentChangeShinsei(Int32)

Declaration

```
public void SetCurrentChangeShinsei(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

UpdateDetailViewVisual()

Declaration

```
public void UpdateDetailViewVisual()
```

Class CardView

Inheritance

System.Object
CardView

Implements

IPointerClickHandler

Namespace: [Timba.SacredTails.CardStoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CardView : MonoBehaviour
```

Fields

OnClickHandler

Declaration

```
public Action<int, CardView> OnClickHandler
```

Field Value

TYPE	DESCRIPTION
Action<System.Int32, CardView >	

Properties

ActionCard

Declaration

```
public ActionCard ActionCard { get; set; }
```

Property Value

TYPE	DESCRIPTION
ActionCard	

Methods

OnPointerClick(PointerEventData)

Declaration

```
public void OnPointerClick(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

UpdateNumber(Int32, Int32)

Declaration

```
public void UpdateNumber(int number, int indexCard)
```


Parameters

TYPE	NAME	DESCRIPTION
System.Int32	number	
System.Int32	indexCard	

Implements

IPointerClickHandler

Class Deck

Inheritance

System.Object

Deck

Namespace: [Timba.SacredTails.CardStoreModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class Deck
```

Fields

cards

Declaration

```
public List<Card> cards
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Card >	

Namespace Timba.SacredTails.CharacterStyle

Classes

[BakeColorsInCharacter](#)

This component allow you to define the color of the NPC in the game

[CharacterRecolor](#)

This component controls the recolor of a character

[CharacterStyleButton](#)

UI element that represents a part of character for edition

[CharacterStyleController](#)

Behavior of panel Character Style Controller

[CharacterStyleDatabase](#)

[PlayerMouseRotator](#)

Allow player rotate character when is on Character Style Panel

Enums

[PartsOfCharacter](#)

Class BakeColorsInCharacter

This component allow you to define the color of the NPC in the game

Inheritance

System.Object

BakeColorsInCharacter

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BakeColorsInCharacter : MonoBehaviour
```

Fields

bodyStyle

Declaration

```
public BodyStyle bodyStyle
```

Field Value

TYPE	DESCRIPTION
BodyStyle	

characterRecolor

Declaration

```
public CharacterRecolor characterRecolor
```

Field Value

TYPE	DESCRIPTION
CharacterRecolor	

Details

Declaration

```
public Color Details
```

Field Value

TYPE	DESCRIPTION
Color	

Hair

Declaration

```
public Color Hair
```

Field Value

TYPE	DESCRIPTION
Color	

HairStyle

Declaration

public int HairStyle

Field Value

TYPE	DESCRIPTION
System.Int32	

materialReskins

Declaration

public List<MaterialReskin> materialReskins

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< MaterialReskin >	

Primary

Declaration

public Color Primary

Field Value

TYPE	DESCRIPTION
Color	

Secondary

Declaration

public Color Secondary

Field Value

TYPE	DESCRIPTION
Color	

Skin

Declaration

public Color Skin

Field Value

TYPE	DESCRIPTION
Color	

Methods

UpdateVisual()

Declaration

```
public void UpdateVisual()
```

Class CharacterRecolor

This component controls the recolor of a character

Inheritance

System.Object

CharacterRecolor

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterRecolor : MonoBehaviour
```

Methods

ChangeMaterialColors(PartsOfCharacter, Color)

Declaration

```
public void ChangeMaterialColors(PartsOfCharacter part, Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	part	
Color	color	

Init(Material)

Declaration

```
public void Init(Material target)
```

Parameters

TYPE	NAME	DESCRIPTION
Material	target	

Class CharacterStyleButton

UI element that represents a part of character for edition

Inheritance

System.Object
CharacterStyleButton

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: [cs.temp.dll.dll](#)

Syntax

```
public class CharacterStyleButton : MonoBehaviour
```

Methods

UpdateBtn(Int32, Boolean)

Declaration

```
public void UpdateBtn(int index, bool enable)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	
System.Boolean	enable	

Class CharacterStyleController

Behavior of panel Character Style Controller

Inheritance

System.Object

CharacterStyleController

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CharacterStyleController : MonoBehaviour
```

Fields

cameraBackgroundColor

Declaration

```
public Color cameraBackgroundColor
```

Field Value

TYPE	DESCRIPTION
Color	

characterRecolor

Declaration

```
public CharacterRecolor characterRecolor
```

Field Value

TYPE	DESCRIPTION
CharacterRecolor	

colorRowPrefab

Declaration

```
public Transform colorRowPrefab
```

Field Value

TYPE	DESCRIPTION
Transform	

colorRowsParent

Declaration

```
public Transform colorRowsParent
```

Field Value

TYPE	DESCRIPTION
Transform	

colorSlotEmptyPrefab

Declaration

public Transform colorSlotEmptyPrefab

Field Value

TYPE	DESCRIPTION
Transform	

colorSlotLockedPrefab

Declaration

public Transform colorSlotLockedPrefab
--

Field Value

TYPE	DESCRIPTION
Transform	

colorSlotUnlockedPrefab

Declaration

public Transform colorSlotUnlockedPrefab
--

Field Value

TYPE	DESCRIPTION
Transform	

layerMaskForStyle

Declaration

public LayerMask layerMaskForStyle

Field Value

TYPE	DESCRIPTION
LayerMask	

lobbyCamera

Declaration

public CinemachineVirtualCamera lobbyCamera

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

numberOfColumnsPerRow

Declaration

public int numberOfColumnsPerRow

Field Value

TYPE	DESCRIPTION
System.Int32	

partRowPrefab

Declaration

public Transform partRowPrefab

Field Value

TYPE	DESCRIPTION
Transform	

partRowsParent

Declaration

public Transform partRowsParent

Field Value

TYPE	DESCRIPTION
Transform	

partSlotLockedPrefab

Declaration

public Transform partSlotLockedPrefab

Field Value

TYPE	DESCRIPTION
Transform	

partSlotUnlockedPrefab

Declaration

public Transform partSlotUnlockedPrefab

Field Value

TYPE	DESCRIPTION
Transform	

playerMouseRotator

Declaration

public PlayerMouseRotator playerMouseRotator
--

Field Value

TYPE	DESCRIPTION
PlayerMouseRotator	

styleCamera

Declaration

public CinemachineVirtualCamera styleCamera

Field Value

TYPE	DESCRIPTION
CinemachineVirtualCamera	

styleDB

Declaration

public CharacterStyleDatabase styleDB

Field Value

TYPE	DESCRIPTION
CharacterStyleDatabase	

title

Declaration

public TMP_Text title

Field Value

TYPE	DESCRIPTION
TMP_Text	

Methods

ApplyLastStyle()

This download the character style data and apply to the model in the game

Declaration

public void ApplyLastStyle()

Awake()

Declaration

```
public void Awake()
```

ChangeHairBodyPart(Int32)

Show a different part of hair in the model

Declaration

```
public void ChangeHairBodyPart(int targetPart)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	targetPart	

ChangeTitle(PartsOfCharacter)

Declaration

```
public string ChangeTitle(PartsOfCharacter part)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	part	

Returns

TYPE	DESCRIPTION
System.String	

GenerateRandomOutfit()

This fill all necessary data for character style with random values

Declaration

```
public void GenerateRandomOutfit()
```

Hide()

Declaration

```
public void Hide()
```

Init()

Declaration

```
public void Init()
```

OnSelectPart(Int32)

Declaration

```
public void OnSelectPart(int partSelected)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	partSelected	

PopulateSkinColors()

Fill panel of color options using a scriptable object

Declaration

```
public void PopulateSkinColors()
```

SelectPartNext(Boolean)

Declaration

```
public void SelectPartNext(bool left)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	left	

ShowCategory(PartsOfCharacter)

Declaration

```
public void ShowCategory(PartsOfCharacter part)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	part	

UpdateCharacterStyle()

Declaration

```
public void UpdateCharacterStyle()
```

UpdateColorPartOfCharacter(PartsOfCharacter, Color, Boolean)

Declaration

```
public static void UpdateColorPartOfCharacter(PartsOfCharacter part, Color color, bool isFill = true)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	part	
Color	color	
System.Boolean	isFill	

UpdateGender()

Toggle between male and female model

Declaration

```
public void UpdateGender()
```

UpdatePartOfCharacter(PartsOfCharacter, Int32)

Declaration

```
public static void UpdatePartOfCharacter(PartsOfCharacter part, int partIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	part	
System.Int32	partIndex	

Class CharacterStyleDatabase

Inheritance

System.Object
CharacterStyleDatabase

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class CharacterStyleDatabase : ScriptableObject
```

Fields

colorDB

Declaration

```
public List<ColorIdRelation> colorDB
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ColorIdRelation >	

partDB

Declaration

```
public List<PartIdRelation> partDB
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< PartIdRelation >	

Methods

GetColorById(Int32, PartsOfCharacter)

Declaration

```
public ColorIdRelation GetColorById(int id, PartsOfCharacter slotType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
PartsOfCharacter	slotType	

Returns

TYPE	DESCRIPTION
ColorIdRelation	

GetColorsByPartType(PartsOfCharacter)

Declaration

```
public List<ColorIdRelation> GetColorsByPartType(PartsOfCharacter slotType)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	slotType	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<ColorIdRelation>	

GetPartById(Int32, PartsOfCharacter)

Declaration

```
public PartIdRelation GetPartById(int id, PartsOfCharacter slotType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
PartsOfCharacter	slotType	

Returns

TYPE	DESCRIPTION
PartIdRelation	

GetPartsByType(PartsOfCharacter)

Declaration

```
public List<PartIdRelation> GetPartsByType(PartsOfCharacter slotType)
```

Parameters

TYPE	NAME	DESCRIPTION
PartsOfCharacter	slotType	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<PartIdRelation>	

OnValidate()

Declaration

```
public void OnValidate()
```

Enum PartsOfCharacter

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: cs.temp.dll.dll

Syntax

```
public enum PartsOfCharacter
```

Fields

NAME	DESCRIPTION
COLORS	
DETAILS	
HAIR	
HANDS	
LEGS	
PRIMARY_COLOR	
SECONDARY_COLOR	
SKIN	

Class PlayerMouseRotator

Allow player rotate character when is on Character Style Panel

Inheritance

System.Object

PlayerMouseRotator

Namespace: [Timba.SacredTails.CharacterStyle](#)

Assembly: [cs.temp.dll.dll](#)

Syntax

```
public class PlayerMouseRotator : MonoBehaviour
```

Fields

canRotateWithMouse

Declaration

```
public bool canRotateWithMouse
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isClickingOnArea

Declaration

```
public bool isClickingOnArea
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

rotationTarget

Declaration

```
public Transform rotationTarget
```

Field Value

TYPE	DESCRIPTION
Transform	

Methods

IsClickingOnArea(Boolean)

Declaration

```
public void IsClickingOnArea(bool isClicking)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isClicking	

Namespace Timba.SacredTails.ChatModule

Classes

[ChatEmojis](#)

[ChatEmojis.DiccionarioOfEmojis](#)

[EmojiButton](#)

[FilterController](#)

Class ChatEmojis

Inheritance

System.Object

ChatEmojis

Namespace: [Timba.SacredTails.ChatModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChatEmojis : MonoBehaviour
```

Fields

chatTextBox

Declaration

```
public ChatTextBox chatTextBox
```

Field Value

TYPE	DESCRIPTION
ChatTextBox	

diccionarioOfEmojis

Declaration

```
public ChatEmojis.DiccionarioOfEmojis diccionarioOfEmojis
```

Field Value

TYPE	DESCRIPTION
ChatEmojis.DiccionarioOfEmojis	

inputField

Declaration

```
public TMP_InputField inputField
```

Field Value

TYPE	DESCRIPTION
TMP_InputField	

pendingCaret

Declaration

```
public int pendingCaret
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

OnChangeValue(String)

Declaration

```
public void OnChangeValue(string msg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	msg	

Update()

Declaration

```
public void Update()
```

Class ChatEmojis.DiccionarioOfEmojis

Inheritance

System.Object

ChatEmojis.DiccionarioOfEmojis

Namespace: [Timba.SacredTails.ChatModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class DiccionarioOfEmojis
```

Fields

keys

Declaration

```
public List<string> keys
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

RealValues

Declaration

```
public List<string> RealValues
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

visual

Declaration

```
public List<Sprite> visual
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

Methods

GetByKey(String)

Declaration

```
public string GetByKey(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Returns

TYPE	DESCRIPTION
System.String	

Class EmojiButton

Inheritance

System.Object

EmojiButton

Namespace: [Timba.SacredTails.ChatModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class EmojiButton : MonoBehaviour
```

Fields

button

Declaration

```
public Button button
```

Field Value

TYPE	DESCRIPTION
Button	

codeValue

Declaration

```
public string codeValue
```

Field Value

TYPE	DESCRIPTION
System.String	

icon

Declaration

```
public Image icon
```

Field Value

TYPE	DESCRIPTION
Image	

realValue

Declaration

```
public string realValue
```

Field Value

TYPE	DESCRIPTION
System.String	

Class FilterController

Inheritance

System.Object

FilterController

Namespace: [Timba.SacredTails.ChatModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class FilterController : MonoBehaviour
```

Fields

chatTextBox

Declaration

```
public ChatTextBox chatTextBox
```

Field Value

TYPE	DESCRIPTION
ChatTextBox	

toggles

Declaration

```
public List<Toggle> toggles
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Toggle>	

Methods

CallApplyFilters()

Declaration

```
public void CallApplyFilters()
```

ChangeAllMinusLast(Boolean)

Declaration

```
public void ChangeAllMinusLast(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

IfAllEnable()

Declaration

```
public void IfAllEnable()
```

IfAnyDisable()

Declaration

```
public void IfAnyDisable()
```

Namespace Timba.SacredTails.Database

Classes

[AuthController](#)

[DatabaseConnection](#)

Interfaces

[IDatabase](#)

Class AuthController

Inheritance

System.Object

AuthController

Namespace: [Timba.SacredTails.Database](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AuthController : MonoBehaviour
```

Fields

ipField

Declaration

```
public TMP_InputField ipField
```

Field Value

TYPE	DESCRIPTION
TMP_InputField	

tournamentCreationLogin

Declaration

```
public bool tournamentCreationLogin
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

transport

Declaration

```
public UnityTransport transport
```

Field Value

TYPE	DESCRIPTION
UnityTransport	

Methods

EventsListenging()

Declaration

```
public void EventsListenging()
```

LoadCachedData()

Declaration

```
public void LoadCachedData()
```

Login()

Declaration

```
public void Login()
```

OpenForgotPasswordPanel(Boolean)

Declaration

```
public void OpenForgotPasswordPanel(bool isLogin)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLogin	

OpenPanel(Boolean)

Declaration

```
public void OpenPanel(bool isLogin)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isLogin	

SendPasswordEmail()

Declaration

```
public void SendPasswordEmail()
```

SetIP()

Declaration

```
public void SetIP()
```

SignUp()

Declaration

```
public void SignUp()
```

Class DatabaseConnection

Inheritance
System.Object
DatabaseConnection

Implements
[IDatabase](#)
IService

Namespace: [Timba.SacredTails.Database](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class DatabaseConnection : MonoBehaviour, IDatabase
```

Methods

CardDatabaseCount()

Declaration

```
public int CardDatabaseCount()
```

Returns

TYPE	DESCRIPTION
System.Int32	

CheckTicketStatus(CreateMatchmakingTicketResult, String)

Declaration

```
public async void CheckTicketStatus(CreateMatchmakingTicketResult ticket, string gamemodeQueue)
```

Parameters

TYPE	NAME	DESCRIPTION
CreateMatchmakingTicketResult	ticket	
System.String	gamemodeQueue	

GetActionCardByIndex(Int32)

Declaration

```
public ActionCard GetActionCardByIndex(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Returns

TYPE	DESCRIPTION
ActionCard	

GetDatabaseCardsIndexListByType(List<String>)

Declaration

```
public List<int> GetDatabaseCardsIndexListByType(List<string> partTypes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	partTypes	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

GetRandomShinsei()

Declaration

```
public string GetRandomShinsei()
```

Returns

TYPE	DESCRIPTION
System.String	

GetSetName(String, String)

Declaration

```
public string GetSetName(string partRarity, string partType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	partRarity	
System.String	partType	

Returns

TYPE	DESCRIPTION
System.String	

GetShinseiPartsTypes(String, Enum)

Declaration

```
public Dictionary<string, string> GetShinseiPartsTypes(string shinseiDna, Enum genericEnum)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	
Enum	genericEnum	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

GetShinseiStats(String)

Declaration

```
public ShinseiStats GetShinseiStats(string shinseiDna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	

Returns

TYPE	DESCRIPTION
ShinseiStats	

GetShinseiStructure(String)

Declaration

```
public string GetShinseiStructure(string dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
System.String	

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

ObtainShinseiRarity(String)

Declaration

```
public RarityType ObtainShinseiRarity(string dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
RarityType	

ObtainShinseiType(String)

Declaration

```
public CharacterType ObtainShinseiType(string dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
CharacterType	

StartMatchmakingSequence(Int32, String)

starts the matchmaking sequence under the specified queue

Declaration

```
public void StartMatchmakingSequence(int skillLevel, string gamemodeQueue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	skillLevel	
System.String	gamemodeQueue	

Implements

[IDatabase](#)

IService

Interface IDatabase

Namespace: [Timba.SacredTails.Database](#)
Assembly: cs.temp.dll.dll

Syntax

```
public interface IDatabase
```

Methods

CardDatabaseCount()

Declaration

```
int CardDatabaseCount()
```

Returns

TYPE	DESCRIPTION
System.Int32	

GetActionCardByIndex(Int32)

Declaration

```
ActionCard GetActionCardByIndex(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Returns

TYPE	DESCRIPTION
ActionCard	

GetDatabaseCardsIndexListByType(List<String>)

Declaration

```
List<int> GetDatabaseCardsIndexListByType(List<string> type)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	type	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

GetRandomShinsei()

Declaration

```
string GetRandomShinsei()
```

Returns

TYPE	DESCRIPTION
System.String	

GetShinseiPartsTypes(String, Enum)

Declaration

```
Dictionary<string, string> GetShinseiPartsTypes(string shinseiDna, Enum genericEnum)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	
Enum	genericEnum	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

GetShinseiStats(String)

Declaration

```
ShinseiStats GetShinseiStats(string shinseiDna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiDna	

Returns

TYPE	DESCRIPTION
ShinseiStats	

GetShinseiStructure(String)

Declaration

```
string GetShinseiStructure(string dna)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
System.String	

ObtainShinseiRarity(String)

Declaration

RarityType ObtainShinseiRarity(string dna)
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
RarityType	

ObtainShinseiType(String)

Declaration

CharacterType ObtainShinseiType(string dna)

Parameters

TYPE	NAME	DESCRIPTION
System.String	dna	

Returns

TYPE	DESCRIPTION
CharacterType	

StartMatchmakingSequence(Int32, String)

Declaration

void StartMatchmakingSequence(int skillLevel = 0, string gamemodeQueue = "FriendlyMatch")

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	skillLevel	
System.String	gamemodeQueue	

Namespace Timba.SacredTails.DialogSystem

Classes

[ActionNode](#)

[AnswerField](#)

[Conversation](#)

This class keep the logic of write dialogs in screen, received a dialog graph and play them

[Dialogable](#)

A component that allow start conversations

[DialogGraph](#)

[DialogNode](#)

[DialogNodeEditor](#)

[DialogUI](#)

Draw necessary components to dialog system in player screen

Structs

[ActionNode.Connection](#)

[DialogNode.Connection](#)

Class ActionNode

Inheritance

System.Object

ActionNode

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ActionNode : Node
```

Fields

callback

Declaration

```
public Action callback
```

Field Value

TYPE	DESCRIPTION
Action	

dialogText

Declaration

```
public string dialogText
```

Field Value

TYPE	DESCRIPTION
System.String	

input

Declaration

```
public ActionNode.Connection input
```

Field Value

TYPE	DESCRIPTION
ActionNode.Connection	

output

Declaration

```
public ActionNode.Connection output
```

Field Value

TYPE	DESCRIPTION
ActionNode.Connection	

Struct ActionNode.Connection

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public struct Connection
```

Class AnswerField

Inheritance

System.Object

AnswerField

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AnswerField : MonoBehaviour
```

Methods

SetButtonResponse(Int32, String, Conversation, Action, Action)

Declaration

```
public void SetButtonResponse(int index, string text, Conversation conversation, Action backButtonCallback = null, Action skipDialog = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	
System.String	text	
Conversation	conversation	
Action	backButtonCallback	
Action	skipDialog	

Class Conversation

This class keep the logic of write dialogs in screen, received a dialog graph and play them

Inheritance

System.Object

Conversation

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Conversation
```

Fields

currentNode

Declaration

```
public DialogNode currentNode
```

Field Value

TYPE	DESCRIPTION
DialogNode	

dialogGraph

Declaration

```
public DialogGraph dialogGraph
```

Field Value

TYPE	DESCRIPTION
DialogGraph	

isPlayerReadText

Declaration

```
public bool isPlayerReadText
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isWaitingResponse

Declaration

```
public bool isWaitingResponse
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

responseIndex

Declaration

public int responseIndex

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

ConversationRoutine(DialogUI, Action, Dialogable)

Declaration

public IEnumerable ConversationRoutine(DialogUI dialogUI, Action EndConversationCallback = null, Dialogable targetDialogable = null)
--

Parameters

TYPE	NAME	DESCRIPTION
DialogUI	dialogUI	
Action	EndConversationCallback	
Dialogable	targetDialogable	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Init(DialogUI, Action)

Declaration

public void Init(DialogUI dialogUI, Action EndConversationCallback)

Parameters

TYPE	NAME	DESCRIPTION
DialogUI	dialogUI	
Action	EndConversationCallback	

NotifyPlayerReadText()

Declaration

public void NotifyPlayerReadText()

SendResponse(Int32)

Declaration

```
public void SendResponse(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

UpdateDialog(DialogNode, DialogUI, Action)

Declaration

```
public void UpdateDialog(DialogNode dialogNode, DialogUI dialogUI, Action EndConversationCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
DialogNode	dialogNode	
DialogUI	dialogUI	
Action	EndConversationCallback	

Class Dialogable

A component that allow start conversations

Inheritance

System.Object

Dialogable

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Dialogable : MonoBehaviour
```

Fields

CallbackEvents

Events that can be triggerred in conversations using a code character

Declaration

```
public List<UnityEvent> CallbackEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<UnityEvent>	

Remarks

You can find a example of use of this in NPC5_Conversation_Out scriptable object at Assets - _SacredTails - Lobby - Scripts - DialogSystem - Conversations\NPC5_Conversation_Out.asset.

dialogUI

Declaration

```
public DialogUI dialogUI
```

Field Value

TYPE	DESCRIPTION
DialogUI	

isExecutingAnswerWithCallbackEvent

Declaration

```
public bool isExecutingAnswerWithCallbackEvent
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EndConversation()

Declaration

```
public void EndConversation()
```

EnteringAnotherPanel()

Declaration

```
public void EnteringAnotherPanel()
```

StartConversation()

This start a new conversation

Declaration

```
public void StartConversation()
```

Class DialogGraph

Inheritance

System.Object

DialogGraph

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DialogGraph : NodeGraph
```


Class DialogNode

Inheritance

System.Object

DialogNode

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DialogNode : Node
```

Fields

Answers

Declaration

```
public List<string> Answers
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

dialogText

Declaration

```
public string dialogText
```

Field Value

TYPE	DESCRIPTION
System.String	

input

Declaration

```
public DialogNode.Connection input
```

Field Value

TYPE	DESCRIPTION
DialogNode.Connection	

output

Declaration

```
public DialogNode.Connection output
```

Field Value

TYPE	DESCRIPTION
DialogNode.Connection	

randomAnswer

Declaration

```
public bool randomAnswer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

sequentialAnswers

Declaration

```
public bool sequentialAnswers
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

sequentialCurrentAnswerKey

Declaration

```
public string sequentialCurrentAnswerKey
```

Field Value

TYPE	DESCRIPTION
System.String	

Struct DialogNode.Connection

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public struct Connection
```

Class DialogNodeEditor

Inheritance

System.Object

DialogNodeEditor

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DialogNodeEditor : NodeEditor
```

Fields

ScrollPos

Declaration

```
public Vector2 ScrollPos
```

Field Value

TYPE	DESCRIPTION
Vector2	

Methods

OnBodyGUI()

Declaration

```
public override void OnBodyGUI()
```

Class DialogUI

Draw necessary components to dialog system in player screen

Inheritance

System.Object

DialogUI

Namespace: [Timba.SacredTails.DialogSystem](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DialogUI : MonoBehaviour
```

Fields

answerFields

Declaration

```
public List<AnswerField> answerFields
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< AnswerField >	

Properties

IsPlayerDialogate

Declaration

```
public bool IsPlayerDialogate { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

ShowResponses(List<String>, Conversation, Action)

Declaration

```
public void ShowResponses(List<string> responses, Conversation conversation, Action EndConversationCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	responses	
Conversation	conversation	
Action	EndConversationCallback	

SkipDialog(Boolean)

Declaration

```
public void SkipDialog(bool forceSkip = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forceSkip	

WriteText(String, Action, Boolean)

Declaration

```
public void WriteText(string dialogText, Action OnEndWrite = null, bool isTextWithAnswer = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dialogText	
Action	OnEndWrite	
System.Boolean	isTextWithAnswer	

Namespace Timba.SacredTails.Interaction

Classes

[SittableMechanic](#)

Allow to player see their character sitdown in chairs on the bar

Class SittableMechanic

Allow to player see their character sitdown in chairs on the bar

Inheritance

System.Object

SittableMechanic

Namespace: [Timba.SacredTails.Interaction](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SittableMechanic : MonoBehaviour
```


Namespace Timba.SacredTails.Lobby

Classes

[PlayerCameraFollow](#)

[ShinseiSpawner](#)

Spawns shinsei as network objects for the lobby scene

Class PlayerCameraFollow

Inheritance

System.Object

Utils.Singleton

Utils.Singleton<[PlayerCameraFollow](#)>

PlayerCameraFollow

Namespace: [Timba.SacredTails.Lobby](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayerCameraFollow : Singleton<PlayerCameraFollow>
```

Fields

virtualCam

Declaration

```
public List<CinemachineVirtualCamera> virtualCam
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<CinemachineVirtualCamera>	

Methods

FollowPlayer(Transform)

Declaration

```
public void FollowPlayer(Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	

Class ShinseiSpawner

Spawns shinsei as network objects for the lobby scene

Inheritance

System.Object

ShinseiSpawner

Namespace: [Timba.SacredTails.Lobby](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiSpawner : MonoBehaviour
```

Fields

characterSlot

Declaration

```
public CharacterSlot characterSlot
```

Field Value

TYPE	DESCRIPTION
CharacterSlot	

Methods

ChangeCurrentShinsei(String)

Declaration

```
public void ChangeCurrentShinsei(string newValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	newValue	

OnSpawn(Transform, Boolean, Boolean)

Declaration

```
public void OnSpawn(Transform parent, bool isLocalPlayerShinsei = false, bool useGravity = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	parent	
System.Boolean	isLocalPlayerShinsei	
System.Boolean	useGravity	

SpawnOtherShinsei(String, Transform, Vector3)

Declaration

```
public void SpawnOtherShinsei(string shinseiCompanionDna, Transform parent, Vector3 playerPos)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	shinseiCompanionDna	
Transform	parent	
Vector3	playerPos	

Namespace Timba.SacredTails.Navigation

Classes

[NavigationManager](#)

Interfaces

[INavigation](#)

Interface INavigation

Namespace: [Timba.SacredTails.Navigation](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface INavigation
```

Methods

GoToScene(Int32)

Declaration

```
void GoToScene(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Class NavigationManager

Inheritance

System.Object

NavigationManager

Implements

[INavigation](#)

IService

Namespace: [Timba.SacredTails.Navigation](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NavigationManager : MonoBehaviour, INavigation
```

Methods

GoToScene(Int32)

Declaration

```
public void GoToScene(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Implements

[INavigation](#)

IService

Namespace Timba.SacredTails.OptionsModule

Classes

[GameSettingsController](#)

Class GameSettingsController

Inheritance

System.Object

GameSettingsController

Namespace: [Timba.SacredTails.OptionsModule](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class GameSettingsController : MonoBehaviour
```

Fields

badWordFilterToggle

Declaration

```
public Toggle badWordFilterToggle
```

Field Value

TYPE	DESCRIPTION
Toggle	

frenchLayoutToggle

Declaration

```
public Toggle frenchLayoutToggle
```

Field Value

TYPE	DESCRIPTION
Toggle	

fullscreenToggle

Declaration

```
public Toggle fullscreenToggle
```

Field Value

TYPE	DESCRIPTION
Toggle	

graphicsDropdown

Declaration

```
public TMP_Dropdown graphicsDropdown
```

Field Value

TYPE	DESCRIPTION
TMP_Dropdown	

qualityLevel

Declaration

```
public RenderPipelineAsset[] qualityLevel
```

Field Value

TYPE	DESCRIPTION
RenderPipelineAsset[]	

resolutionDropdown

Declaration

```
public TMP_Dropdown resolutionDropdown
```

Field Value

TYPE	DESCRIPTION
TMP_Dropdown	

settingsContainer

Declaration

```
public GameObject settingsContainer
```

Field Value

TYPE	DESCRIPTION
GameObject	

settingsPanel

Declaration

```
public GameObject settingsPanel
```

Field Value

TYPE	DESCRIPTION
GameObject	

terrain

Declaration

```
public Terrain terrain
```

Field Value

TYPE	DESCRIPTION
Terrain	

versionText

Declaration

```
public TMP_Text versionText
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Methods

ChangeBadWordFilter(Boolean)

Declaration

```
public void ChangeBadWordFilter(bool changeToogle)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	changeToogle	

ChangeFullScreen()

Declaration

```
public void ChangeFullScreen()
```

ChangeFullScreen(Boolean)

Declaration

```
public void ChangeFullScreen(bool changeToogle)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	changeToogle	

ChangeGraphicSettings(Int32)

Declaration

```
public void ChangeGraphicSettings(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	

ChangeKeyboardLayout(Boolean)

Declaration

```
public void ChangeKeyboardLayout(bool changeToogle)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	changeToogle	

ChangeResolution(Int32)

Declaration

```
public void ChangeResolution(int resolutionSettings)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	resolutionSettings	

GetResolution(Int32, Boolean)

Declaration

```
public Vector2Int GetResolution(int option, bool saveOnPlayerPrefs = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	option	
System.Boolean	saveOnPlayerPrefs	

Returns

TYPE	DESCRIPTION
Vector2Int	

SettingsButtonInteract(Boolean)

Declaration

```
public void SettingsButtonInteract(bool active)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	active	

Start()

Declaration

```
public void Start()
```

Namespace Timba.SacredTails.Photoboot

Classes

[ShinseiWardrobe](#)

This allow take pictures to the Shinsei, using Camera and Coroutines

Interfaces

[IconGeneration](#)

Service that allows take pictures of Shinsei for UI

Interface IIconGeneration

Service that allows take pictures of Shinsei for UI

Namespace: [Timba.SacredTails.Photoboot](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface IIconGeneration
```

Methods

GenerateShinseiIcons(List<Shinsei>, Action)

Declaration

```
void GenerateShinseiIcons(List<Shinsei> shinseiParty, Action callback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< Shinsei >	shinseiParty	
Action	callback	

Class ShinseiWardrobe

This allow take pictures to the Shinsei, using Camera and Coroutines

Inheritance

System.Object
ShinseiWardrobe

Implements

[IIconGeneration](#)
IService

Namespace: [Timba.SacredTails.Photoboot](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class ShinseiWardrobe : MonoBehaviour, IIconGeneration
```

Fields

generatedSprite

Declaration

```
public Sprite generatedSprite
```

Field Value

TYPE	DESCRIPTION
Sprite	

Methods

GenerateShinseiIcons(List<Shinsei>, Action)

Declaration

```
public void GenerateShinseiIcons(List<Shinsei> shinseiParty, Action callback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< Shinsei >	shinseiParty	
Action	callback	

GetGeneratedIcons(List<Shinsei>)

Declaration

```
public List<Sprite> GetGeneratedIcons(List<Shinsei> shinseiParty)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< Shinsei >	shinseiParty	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

GetGeneratedSequence(List<Shinsei>, Int32)

Declaration

```
public List<Sprite> GetGeneratedSequence(List<Shinsei> shinseiParty, int targetFrames)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< Shinsei >	shinseiParty	
System.Int32	targetFrames	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<Sprite>	

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

TexToSprite(Texture2D)

Declaration

```
public Sprite TexToSprite(Texture2D tex)
```

Parameters

TYPE	NAME	DESCRIPTION
Texture2D	tex	

Returns

TYPE	DESCRIPTION
Sprite	

Implements

[IIconGeneration](#)

[IService](#)

Namespace Timba.SacredTails.TournamentBehavior

Classes

[SearchAndShow](#)

Show all active tournaments registered in the game

[SingleBracketPlayerLabelController](#)

[TournamentBracketsShowController](#)

[TournamentCreationController](#)

[TournamentEntry](#)

[TournamentReadyController](#)

[TournamentSlot](#)

UI Element of clickable button for join to specific tournament

[TournamentSlot.JoinTournamentDto](#)

[TournamentSlot.TournamentPlayerDataDto](#)

Class SearchAndShow

Show all active tournaments registered in the game

Inheritance

System.Object

SearchAndShow

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SearchAndShow : MonoBehaviour
```

Fields

tournamentSlots

Declaration

```
public List<TournamentSlot> tournamentSlots
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TournamentSlot >	

Methods

SearchAndShowTournaments()

Declaration

```
public void SearchAndShowTournaments()
```

Class SingleBracketPlayerLabelController

Inheritance

System.Object

SingleBracketPlayerLabelController

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SingleBracketPlayerLabelController : MonoBehaviour
```

Fields

bracketLineBack

Declaration

```
public Image bracketLineBack
```

Field Value

TYPE	DESCRIPTION
Image	

bracketLineDown

Declaration

```
public Image bracketLineDown
```

Field Value

TYPE	DESCRIPTION
Image	

bracketLineFront

Declaration

```
public Image bracketLineFront
```

Field Value

TYPE	DESCRIPTION
Image	

bracketLineGlowin

Declaration

```
public Sprite bracketLineGlowin
```

Field Value

TYPE	DESCRIPTION
Sprite	

bracketLineUp

Declaration

```
public Image bracketLineUp
```

Field Value

TYPE	DESCRIPTION
Image	

labelFrameReference

Declaration

```
public Image labelFrameReference
```

Field Value

TYPE	DESCRIPTION
Image	

labelFrameWhite

Declaration

```
public Sprite labelFrameWhite
```

Field Value

TYPE	DESCRIPTION
Sprite	

playerDisplayName

Declaration

```
public TMP_Text playerDisplayName
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

shinseiIcon

Declaration

```
public Image shinseiIcon
```

Field Value

TYPE	DESCRIPTION
Image	

vslImage

Declaration

```
public Image vsImage
```

Field Value

TYPE	DESCRIPTION
Image	

Methods

InitPlayerLabel(Boolean, Boolean, Boolean, String, Int32, Int32, Boolean, Boolean)

Declaration

```
public void InitPlayerLabel(bool isInitBracket, bool isWinnerBracket, bool isOddNumber, string playerName, int currentStage, int totalStages, bool isWinnerOfPreviousStage, bool isAnExistingStage)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isInitBracket	
System.Boolean	isWinnerBracket	
System.Boolean	isOddNumber	
System.String	playerName	
System.Int32	currentStage	
System.Int32	totalStages	
System.Boolean	isWinnerOfPreviousStage	
System.Boolean	isAnExistingStage	

Class TournamentBracketsShowController

Inheritance

System.Object

TournamentBracketsShowController

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentBracketsShowController : MonoBehaviour
```

Fields

bracketColumnPrefab

Declaration

```
public GameObject bracketColumnPrefab
```

Field Value

TYPE	DESCRIPTION
GameObject	

bracketLabelPrefab

Declaration

```
public SingleBracketPlayerLabelController bracketLabelPrefab
```

Field Value

TYPE	DESCRIPTION
SingleBracketPlayerLabelController	

bracketsContentParent

Declaration

```
public Transform bracketsContentParent
```

Field Value

TYPE	DESCRIPTION
Transform	

testBracket

Declaration

```
public string testBracket
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

CreateBracketLabel(String, Boolean, Boolean, Int32, GameObject, Int32, Int32, Int32, Boolean, List<String>)

Declaration

```
public void CreateBracketLabel(string _PID, bool _isOddNumber, bool _isInitBracket, int _maxPlayers,
GameObject _currentColumnBracket, int _counterBrackets, int _currentStage, int maxStage, bool
_isAnExistingStage = false, List<string> winnersOfCurrentStage = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_PID	
System.Boolean	_isOddNumber	
System.Boolean	_isInitBracket	
System.Int32	_maxPlayers	
GameObject	_currentColumnBracket	
System.Int32	_counterBrackets	
System.Int32	_currentStage	
System.Int32	maxStage	
System.Boolean	_isAnExistingStage	
System.Collections.Generic.List<System.String>	winnersOfCurrentStage	

FillFinalBracket(Int32, Boolean, Int32, Int32, Int32)

Declaration

```
public void FillFinalBracket(int maxPlayers, bool isInitBracket, int counterBrackets, int currentStage, int
totalStages)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maxPlayers	
System.Boolean	isInitBracket	
System.Int32	counterBrackets	
System.Int32	currentStage	
System.Int32	totalStages	

FillFutureBrackets(Int32, Int32, Int32, ref Boolean, ref Int32)

Declaration

```
public void FillFutureBrackets(int maxPlayers, int totalStages, int currentStage, ref bool isInitBracket, ref int counterBrackets)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maxPlayers	
System.Int32	totalStages	
System.Int32	currentStage	
System.Boolean	isInitBracket	
System.Int32	counterBrackets	

GetTotalNumberOfStagesAndBrackets(Int32)

Declaration

```
public (int, int) GetTotalNumberOfStagesAndBrackets(int maxPlayers)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maxPlayers	

Returns

TYPE	DESCRIPTION
System.ValueTuple<System.Int32, System.Int32>	

GetWinnersOfThisStage(Dictionary<String, StageBracketData>, Int32)

Declaration

```
public List<string> GetWinnersOfThisStage(Dictionary<string, StageBracketData> stagesData, int currentStageIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, StageBracketData >	stagesData	
System.Int32	currentStageIndex	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

InitBracket()

Declaration


```
public void InitBracket()
```

OnEnable()

Declaration

```
public void OnEnable()
```

SetCurrentBrackets(Int32, Int32, Dictionary<String, StageBracketData>, ref Boolean)

Declaration

```
public (int, int) SetCurrentBrackets(int maxPlayers, int totalStages, Dictionary<string, StageBracketData> stagesData, ref bool isInitBracket)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maxPlayers	
System.Int32	totalStages	
System.Collections.Generic.Dictionary<System.String, StageBracketData >	stagesData	
System.Boolean	isInitBracket	

Returns

TYPE	DESCRIPTION
System.ValueTuple<System.Int32, System.Int32>	

SetSpacingBetweenBrackets(VerticalLayoutGroup, Int32)

Declaration

```
public void SetSpacingBetweenBrackets(VerticalLayoutGroup currentColumnBracket, int currentStage)
```

Parameters

TYPE	NAME	DESCRIPTION
VerticalLayoutGroup	currentColumnBracket	
System.Int32	currentStage	

ShowTestTournament()

Declaration

```
public void ShowTestTournament()
```

Class TournamentCreationController

Inheritance

System.Object

TournamentCreationController

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentCreationController : MonoBehaviour
```

Fields

calendarController

Declaration

```
public CalendarController calendarController
```

Field Value

TYPE	DESCRIPTION
CalendarController	

loadingScreen

Declaration

```
public GameObject loadingScreen
```

Field Value

TYPE	DESCRIPTION
GameObject	

maxPlayerInput

Declaration

```
public TMP_Dropdown maxPlayerInput
```

Field Value

TYPE	DESCRIPTION
TMP_Dropdown	

onLoginSuccess

Declaration

```
public UnityEvent onLoginSuccess
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

tournamentName

Declaration

```
public TMP_InputField tournamentName
```

Field Value

TYPE	DESCRIPTION
TMP_InputField	

Methods

CheckIfUserIsAllowed(LoginResult)

Declaration

```
public void CheckIfUserIsAllowed(LoginResult resultLogin)
```

Parameters

TYPE	NAME	DESCRIPTION
LoginResult	resultLogin	

CreateTournament()

Declaration

```
public void CreateTournament()
```

GenerateTournamentId()

Declaration

```
public string GenerateTournamentId()
```

Returns

TYPE	DESCRIPTION
System.String	

Class TournamentEntry

Inheritance

System.Object

TournamentEntry

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentEntry
```

Fields

initTimeStage_1

Declaration

```
public string initTimeStage_1
```

Field Value

TYPE	DESCRIPTION
System.String	

maxPlayer

Declaration

```
public string maxPlayer
```

Field Value

TYPE	DESCRIPTION
System.String	

tournamentId

Declaration

```
public string tournamentId
```

Field Value

TYPE	DESCRIPTION
System.String	

tournamentName

Declaration

```
public string tournamentName
```

Field Value

TYPE	DESCRIPTION
System.String	

Class TournamentReadyController

Inheritance

System.Object

TournamentReadyController

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentReadyController : MonoBehaviour
```

Fields

currentStageTime

Declaration

```
public DateTime currentStageTime
```

Field Value

TYPE	DESCRIPTION
DateTime	

hasCreatedMatch

Declaration

```
public bool hasCreatedMatch
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

labelTimer

Declaration

```
public TMP_Text labelTimer
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

readyButton

Declaration

```
public Button readyButton
```

Field Value

TYPE	DESCRIPTION
Button	

timerLeft

Declaration

```
public float timerLeft
```

Field Value

TYPE	DESCRIPTION
System.Single	

timerText

Declaration

```
public TMP_Text timerText
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Methods

CheckIfWinByDefault(BracketsData, SacredTailsPSDto<BracketsData>)

Declaration

```
public bool CheckIfWinByDefault(BracketsData bracketsData, SacredTailsPSDto<BracketsData> response)
```

Parameters

TYPE	NAME	DESCRIPTION
BracketsData	bracketsData	
SacredTailsPSDto< BracketsData >	response	

Returns

TYPE	DESCRIPTION
System.Boolean	

CreateMatch(BracketsData)

Declaration

```
public void CreateMatch(BracketsData _bracketsData)
```

Parameters

TYPE	NAME	DESCRIPTION
BracketsData	_bracketsData	

EnableReadyButton(Boolean)

Declaration

```
public void EnableReadyButton(bool isReadyEnable)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isReadyEnable	

Enter()

Declaration

```
public void Enter()
```

ExitTournament(String, Action)

Declaration

```
public void ExitTournament(string message = null, Action finishCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	
Action	finishCallback	

GetTimerLeftForNextStage(String)

Set current timer to time left for next round

Declaration

```
public void GetTimerLeftForNextStage(string nextStageTime)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	nextStageTime	

MarkAsReady(Boolean)

Declaration

```
public void MarkAsReady(bool isMarkingReady = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isMarkingReady	

OnDisable()

Declaration

```
public void OnDisable()
```

OnEnable()

Declaration

```
public void OnEnable()
```

ShowPendingTimeAfterMatch(String)

Declaration

```
public void ShowPendingTimeAfterMatch(string nextStageTime = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	nextStageTime	

ShowTimer(Boolean, Action)

Declaration

```
public void ShowTimer(bool showHour, Action onEndTimer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	showHour	
Action	onEndTimer	

ShowTimer(Single, Boolean)

Declaration

```
public void ShowTimer(float timerLeft, bool showHour)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	timerLeft	
System.Boolean	showHour	

ShowTimerCoroutine(Boolean, Action)

Declaration

```
public IEnumerator ShowTimerCoroutine(bool showHour, Action onEndTimer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	showHour	
Action	onEndTimer	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

ShowTimerInitTournament(DateTime)

Declaration

```
public void ShowTimerInitTournament(DateTime initTimeTournament)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	initTimeTournament	

StartReadyButton(Single)

Declaration

```
public void StartReadyButton(float _timerLeft = -1F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	_timerLeft	

Class TournamentSlot

UI Element of clickable button for join to specific tournament

Inheritance

System.Object

TournamentSlot

Implements

IPointerClickHandler

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentSlot : MonoBehaviour
```

Fields

dayName

Declaration

```
public TextMeshProUGUI dayName
```

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

hour

Declaration

```
public TextMeshProUGUI hour
```

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

initTime

Declaration

```
public TextMeshProUGUI initTime
```

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

joinBtn

Declaration

```
public Button joinBtn
```

Field Value

TYPE	DESCRIPTION
Button	

maxPlayers

Declaration

public TextMeshProUGUI maxPlayers

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

OnClick

Declaration

public Action<TournamentSlot, string> OnClick

Field Value

TYPE	DESCRIPTION
Action< TournamentSlot , System.String>	

register

Declaration

public TextMeshProUGUI register

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

tournamentTitle

Declaration

public TextMeshProUGUI tournamentTitle
--

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

Methods

DrawEntry(TournamentEntry)

Declaration

public void DrawEntry(TournamentEntry tournamentEntry)
--

Parameters

TYPE	NAME	DESCRIPTION
TournamentEntry	tournamentEntry	

JoinTournament(String, DateTime, String)

Declaration

```
public void JoinTournament(string tournamentKey, DateTime initTimeTournament, string tournamentName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	tournamentKey	
DateTime	initTimeTournament	
System.String	tournamentName	

OnPointerClick(PointerEventData)

Declaration

```
public void OnPointerClick(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

ShowSelected()

Declaration

```
public void ShowSelected()
```

ShowUnselected()

Declaration

```
public void ShowUnselected()
```

Implements

IPointerClickHandler

Class TournamentSlot.JoinTournamentDto

Inheritance

System.Object

TournamentSlot.JoinTournamentDto

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class JoinTournamentDto
```

Fields

displayName

Declaration

```
public string displayName
```

Field Value

TYPE	DESCRIPTION
System.String	

tournamentPlayerDataDto

Declaration

```
public TournamentSlot.TournamentPlayerDataDto tournamentPlayerDataDto
```

Field Value

TYPE	DESCRIPTION
TournamentSlot.TournamentPlayerDataDto	

Class TournamentSlot.TournamentPlayerDataDto

Inheritance

System.Object

TournamentSlot.TournamentPlayerDataDto

Namespace: [Timba.SacredTails.TournamentBehavior](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class TournamentPlayerDataDto
```

Fields

currentBracket

Declaration

```
public int currentBracket
```

Field Value

TYPE	DESCRIPTION
System.Int32	

displayName

Declaration

```
public string displayName
```

Field Value

TYPE	DESCRIPTION
System.String	

lastPetitionTimeStamp

Declaration

```
public string lastPetitionTimeStamp
```

Field Value

TYPE	DESCRIPTION
System.String	

stage

Declaration

```
public int stage
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Namespace Timba.SacredTails.UiHelpers

Classes

[BackgroundTypeSwapper](#)

[CalendarController](#)

[CalendarDateItem](#)

[ChangeSelectedWithTab](#)

[EntryController](#)

[IconContainerSO](#)

[IconSet](#)

[KeepScrollBar](#)

[LeaderboardElement](#)

[LeaderboardElement.ElementData](#)

[PositionCounter](#)

[ScrollButton](#)

[ScrollViewSystem](#)

[SeekMouseIntoScreen](#)

[SelectableButtonGroup](#)

[SelectableButtonGroup.ButtonList](#)

[TypeIconAsigner](#)

[UIGroups](#)

[UIGroups.UiGroup](#)

[WoldPositionateUiElement](#)

[WoldPositionUiHandler](#)

This allow you to put UI elements attached to player view

Interfaces

[IUIHelpable](#)

Class BackgroundTypeSwapper

Inheritance

System.Object

BackgroundTypeSwapper

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class BackgroundTypeSwapper : MonoBehaviour
```

Methods

CallByShinseiType(Image, CharacterType)

Declaration

```
public void CallByShinseiType(Image image, CharacterType shinseiType)
```

Parameters

TYPE	NAME	DESCRIPTION
Image	image	
CharacterType	shinseiType	

SetImageSpriteByType(Image, Int32)

Declaration

```
public void SetImageSpriteByType(Image image, int indexSprite)
```

Parameters

TYPE	NAME	DESCRIPTION
Image	image	
System.Int32	indexSprite	

Class CalendarController

Inheritance

System.Object

CalendarController

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CalendarController : MonoBehaviour
```

Fields

monthNumText

Declaration

```
public TMP_Text _monthNumText
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

yearNumText

Declaration

```
public TMP_Text _yearNumText
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

dateItems

Declaration

```
public List<CalendarDateItem> dateItems
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< CalendarDateItem >	

hour

Declaration

```
public TMP_Dropdown hour
```

Field Value

TYPE	DESCRIPTION
TMP_Dropdown	

itemPrefab

Declaration

```
public CalendarDateItem itemPrefab
```

Field Value

TYPE	DESCRIPTION
CalendarDateItem	

minute

Declaration

```
public TMP_Dropdown minute
```

Field Value

TYPE	DESCRIPTION
TMP_Dropdown	

onSelectDate

Declaration

```
public UnityEvent onSelectDate
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

second

Declaration

```
public TMP_Dropdown second
```

Field Value

TYPE	DESCRIPTION
TMP_Dropdown	

target

Declaration

```
public TMP_Text target
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Methods

CalendarHasDate()

Declaration

```
public bool CalendarHasDate()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

CreateDateForTournament()

Declaration

```
public string CreateDateForTournament()
```

Returns

TYPE	DESCRIPTION
System.String	

MonthNext()

Declaration

```
public void MonthNext()
```

MonthPrev()

Declaration

```
public void MonthPrev()
```

OnDateItemClick(String)

Declaration

```
public void OnDateItemClick(string day)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	day	

ParseDateNumberToCorrectFormat(String)

Declaration

```
public string ParseDateNumberToCorrectFormat(string number)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	number	

Returns

TYPE	DESCRIPTION
System.String	

YearNext()

Declaration

```
public void YearNext()
```

YearPrev()

Declaration

```
public void YearPrev()
```

Class CalendarDateItem

Inheritance

System.Object

CalendarDateItem

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class CalendarDateItem : MonoBehaviour
```

Fields

onClickDate

Declaration

```
public Action<string> onClickDate
```

Field Value

TYPE	DESCRIPTION
Action<System.String>	

textField

Declaration

```
public TMP_Text textField
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

Methods

OnDateItemClick()

Declaration

```
public void OnDateItemClick()
```

Class ChangeSelectedWithTab

Inheritance

System.Object

ChangeSelectedWithTab

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ChangeSelectedWithTab : MonoBehaviour
```

Class EntryController

Inheritance

System.Object

EntryController

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class EntryController : MonoBehaviour
```

Fields

nft_amount

Declaration

```
public TMP_Text nft_amount
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

nft_image

Declaration

```
public Image nft_image
```

Field Value

TYPE	DESCRIPTION
Image	

qualificator

Declaration

```
public TMP_Text qualificator
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

sc_amount

Declaration

```
public TMP_Text sc_amount
```

Field Value

TYPE	DESCRIPTION
TMP_Text	

sc_image

Declaration

```
public Image sc_image
```

Field Value

TYPE	DESCRIPTION
Image	

Methods

FillData(RankRewardEntry)

Declaration

```
public void FillData(RankRewardEntry data)
```

Parameters

TYPE	NAME	DESCRIPTION
RankRewardEntry	data	

Class IconContainerSO

Inheritance

System.Object

IconContainerSO

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class IconContainerSO : ScriptableObject
```

Fields

IconCollection

Declaration

```
public List<IconSet> IconCollection
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< IconSet >	

nullIcon

Declaration

```
public IconSet nullIcon
```

Field Value

TYPE	DESCRIPTION
IconSet	

Methods

AssignIcon(Nullable<CharacterType>)

Declaration

```
public IconSet AssignIcon(CharacterType? charType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<CharacterType>	charType	

Returns

TYPE	DESCRIPTION
IconSet	

Class IconSet

Inheritance

System.Object

IconSet

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class IconSet
```

Fields

backgroundSprite

Declaration

```
public Sprite backgroundSprite
```

Field Value

TYPE	DESCRIPTION
Sprite	

battleIcon

Declaration

```
public Sprite battleIcon
```

Field Value

TYPE	DESCRIPTION
Sprite	

iconType

Declaration

```
public CharacterType iconType
```

Field Value

TYPE	DESCRIPTION
CharacterType	

negativeIcon

Declaration

```
public Sprite negativeIcon
```

Field Value

TYPE	DESCRIPTION
Sprite	

partIcon

Declaration

```
public Sprite partIcon
```

Field Value

TYPE	DESCRIPTION
Sprite	

TypeColor

Declaration

```
public Color TypeColor
```

Field Value

TYPE	DESCRIPTION
Color	

Interface IUIHelpable

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public interface IUIHelpable
```

Methods

AssignIcon(Nullable<CharacterType>)

Declaration

```
IconSet AssignIcon(CharacterType? charType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<CharacterType>	charType	

Returns

TYPE	DESCRIPTION
IconSet	

Class KeepScrollBar

Inheritance

System.Object

KeepScrollBar

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class KeepScrollBar : MonoBehaviour
```

Methods

setFixedHandleSize()

Declaration

```
public void setFixedHandleSize()
```

Class LeaderboardElement

Inheritance

System.Object

LeaderboardElement

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class LeaderboardElement : MonoBehaviour
```

Fields

button

Declaration

```
public Button button
```

Field Value

TYPE	DESCRIPTION
Button	

Properties

ElementDataObject

Declaration

```
public LeaderboardElement.ElementData ElementDataObject { get; }
```

Property Value

TYPE	DESCRIPTION
LeaderboardElement.ElementData	

Methods

ChangePosition(Int32)

Declaration

```
public void ChangePosition(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	

ChangeVictories(String)

Declaration

```
public void ChangeVictories(string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	

DrawElement(LeaderboardElement.ElementData)

Declaration

```
public void DrawElement(LeaderboardElement.ElementData elementData)
```

Parameters

TYPE	NAME	DESCRIPTION
LeaderboardElement.ElementData	elementData	

Class LeaderboardElement.ElementData

Inheritance

System.Object

LeaderboardElement.ElementData

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ElementData
```

Fields

name

Declaration

```
public string name
```

Field Value

TYPE	DESCRIPTION
System.String	

points

Declaration

```
public string points
```

Field Value

TYPE	DESCRIPTION
System.String	

position

Declaration

```
public string position
```

Field Value

TYPE	DESCRIPTION
System.String	

victories

Declaration

```
public string victories
```

Field Value

TYPE	DESCRIPTION
System.String	

Class PositionCounter

Inheritance

System.Object

PositionCounter

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PositionCounter : MonoBehaviour
```

Methods

EnablePosition(Int32)

Declaration

```
public void EnablePosition(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Class ScrollButton

Inheritance

System.Object
ScrollButton

Implements

IPointerDownHandler
IPointerUpHandler

Namespace: [Timba.SacredTails.UiHelpers](#)
Assembly: cs.temp.dll.dll

Syntax

```
public class ScrollButton : MonoBehaviour
```

Fields

isDown

Declaration

```
public bool isDown
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

OnPointerDown(PointerEventData)

Declaration

```
public void OnPointerDown(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

OnPointerUp(PointerEventData)

Declaration

```
public void OnPointerUp(PointerEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
PointerEventData	eventData	

Implements

IPointerDownHandler
IPointerUpHandler

Class ScrollViewSystem

Inheritance

System.Object

ScrollViewSystem

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ScrollViewSystem : MonoBehaviour
```

Methods

LeftScroll()

Declaration

```
public void LeftScroll()
```

RightScroll()

Declaration

```
public void RightScroll()
```

Start()

Declaration

```
public void Start()
```

Class SeekMouseIntoScreen

Inheritance

System.Object

SeekMouseIntoScreen

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SeekMouseIntoScreen : MonoBehaviour
```

Fields

offsetX

Declaration

```
public int offsetX
```

Field Value

TYPE	DESCRIPTION
System.Int32	

offsetY

Declaration

```
public int offsetY
```

Field Value

TYPE	DESCRIPTION
System.Int32	

posOffsetX

Declaration

```
public int posOffsetX
```

Field Value

TYPE	DESCRIPTION
System.Int32	

posOffsetY

Declaration

```
public int posOffsetY
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

Enable(Boolean)

Declaration

```
public void Enable(bool state)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	state	

Class SelectableButtonGroup

Inheritance

System.Object
SelectableButtonGroup

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class SelectableButtonGroup : MonoBehaviour
```

Methods

SelectButton(Int32)

Declaration

```
public void SelectButton(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Class SelectableButtonGroup.ButtonList

Inheritance

System.Object

SelectableButtonGroup.ButtonList

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class ButtonList
```

Fields

buttonText

Declaration

```
public TextMeshProUGUI buttonText
```

Field Value

TYPE	DESCRIPTION
TextMeshProUGUI	

marker

Declaration

```
public GameObject marker
```

Field Value

TYPE	DESCRIPTION
GameObject	

Class TypeIconAsigner

Inheritance

System.Object
TypeIconAsigner

Implements

IUIHelpable
IService

Namespace: Timba.SacredTails.UiHelpers

Assembly: cs.temp.dll.dll

Syntax

```
public class TypeIconAsigner : MonoBehaviour, IUIHelpable
```

Fields

IconDatabase

Declaration

```
public IconContainerSO IconDatabase
```

Field Value

TYPE	DESCRIPTION
IconContainerSO	

Methods

AssignIcon(Nullable<CharacterType>)

Declaration

```
public IconSet AssignIcon(CharacterType? charType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<CharacterType>	charType	

Returns

TYPE	DESCRIPTION
IconSet	

IsReady()

Declaration

```
public bool IsReady()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Implements

[IUHelpable](#)

IService

Class UIGroups

Inheritance

System.Object

UIGroups

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class UIGroups : MonoBehaviour
```

Fields

groups

Declaration

```
public List<UIGroups.UiGroup> groups
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< UIGroups.UiGroup >	

instance

Declaration

```
public static UIGroups instance
```

Field Value

TYPE	DESCRIPTION
UIGroups	

lastActivate

Declaration

```
public string lastActivate
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

NotifyDynamicPanel(CanvasGroup, String)

Declaration

```
public void NotifyDynamicPanel(CanvasGroup targetCanvas, string key)
```

Parameters

TYPE	NAME	DESCRIPTION
CanvasGroup	targetCanvas	
System.String	key	

ShowOnlyThisGroup(String)

Declaration

```
public void ShowOnlyThisGroup(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

ShowOnlyThisGroupWithDeactivating(String)

Declaration

```
public void ShowOnlyThisGroupWithDeactivating(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Class UIGroups.UiGroup

Inheritance

System.Object

UIGroups.UiGroup

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class UiGroup
```

Fields

canvasOfGroup

Declaration

```
public List<CanvasGroup> canvasOfGroup
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<CanvasGroup>	

key

Declaration

```
public string key
```

Field Value

TYPE	DESCRIPTION
System.String	

Class WoldPositionateUiElement

Inheritance

System.Object
WoldPositionateUiElement

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WoldPositionateUiElement : MonoBehaviour
```

Fields

uiElement

Declaration

```
public GameObject uiElement
```

Field Value

TYPE	DESCRIPTION
GameObject	

Class WoldPositionUIHandler

This allow you to put UI elements attached to player view

Inheritance

System.Object

WoldPositionUIHandler

Namespace: [Timba.SacredTails.UiHelpers](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class WoldPositionUIHandler : MonoBehaviour
```

Fields

instance

Declaration

```
public static WoldPositionUIHandler instance
```

Field Value

TYPE	DESCRIPTION
WoldPositionUIHandler	

WorldPositionateUiElements

Declaration

```
public List<WoldPositionateUiElement> WorldPositionateUiElements
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< WoldPositionateUiElement >	

Methods

RegisterUiElement(WoldPositionateUiElement)

Declaration

```
public void RegisterUiElement(WoldPositionateUiElement target)
```

Parameters

TYPE	NAME	DESCRIPTION
WoldPositionateUiElement	target	

UnregisterUiElement(WoldPositionateUiElement)

Declaration

```
public void UnregisterUiElement(WoldPositionateUiElement target)
```

Parameters

TYPE	NAME	DESCRIPTION
WoldPositionateUiElement	target	

Namespace Timba.SacredTails.VFXController

Classes

[VFXDestroyer](#)

[VFXInstancer](#)

Class VFXDestroyer

Inheritance

System.Object

VFXDestroyer

Namespace: [Timba.SacredTails.VFXController](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VFXDestroyer : MonoBehaviour
```

Class VFXInstancer

Inheritance

System.Object

VFXInstancer

Namespace: [Timba.SacredTails.VFXController](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class VFXInstancer : MonoBehaviour
```

Methods

GetVfx(Int32)

Declaration

```
public VfxInfo GetVfx(int vfxIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	vfxIndex	

Returns

TYPE	DESCRIPTION
VfxInfo	

GetVfxIsReversed(Int32)

Declaration

```
public bool GetVfxIsReversed(int vfxIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	vfxIndex	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetVfxIsReversed(VfxInfo)

Declaration

```
public bool GetVfxIsReversed(VfxInfo vfxInfo)
```

Parameters

TYPE	NAME	DESCRIPTION
VfxInfo	vfxInfo	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetVfxTime(Int32)

Declaration

```
public float GetVfxTime(int vfxIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	vfxIndex	

Returns

TYPE	DESCRIPTION
System.Single	

GetVfxTime(VfxInfo)

Declaration

```
public float GetVfxTime(VfxInfo vfxInfo)
```

Parameters

TYPE	NAME	DESCRIPTION
VfxInfo	vfxInfo	

Returns

TYPE	DESCRIPTION
System.Single	

SpawnVFX(Int32, Vector3, Quaternion)

Declaration

```
public GameObject SpawnVFX(int vfxIndex, Vector3 position, Quaternion rotation)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	vfxIndex	
Vector3	position	

TYPE	NAME	DESCRIPTION
Quaternion	rotation	

Returns

TYPE	DESCRIPTION
GameObject	

Test()

Declaration

```
public void Test()
```

Test2()

Declaration

```
public void Test2()
```