oxygendoc

Generated by Doxygen 1.13.2

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 oxygen Namespace Reference

Namespaces

- namespace AudioAbstraction
- namespace BSPDefines
- namespace GfxSoftwareRasterizer
- namespace GraphicsAbstraction
- namespace InputAbstraction
- namespace Math
- namespace NetworkAbstraction

Classes

- struct AnimatedMeshComponent
- struct AnimatedMeshResource
- struct AnimationInfo
- struct BSPWorldData
- struct CallbackList
- struct CameraComponent
- struct Component
- struct Entity
- struct EntityHierarchy
- struct EnvPushComponent
- struct GameManager
- struct GfxRenderer
- struct GfxTexture
- struct GfxTri
- struct GfxVertex
- struct HealthComponent
- struct HullComponent
- struct InputManager
- struct InternalEngineSingletonsOrder
- struct InternalFileMapWinX64
- struct ManagedObject
- struct NetConnection

- struct NetSystem
- struct NonCopyable

  A non-copyable type to be inherited from. Expresses clearly that a type cannot be copied.
- struct Object
- struct ObjectDescription
- struct ObjectManager
- struct Pawn
- struct ProjectileComponent
- struct ResourceManager
- struct SingletonBase

  Singleton base class. All singletons should inherit from this class. Uses the curiously recurring template pattern and provides the GetInstance() static method to access the singleton.
- struct SingletonHolder

  Storage of a group of singletons. Used to explicitly construct and destruct singletons.
- struct SingletonInstance

  Singleton instance, should only be in an object templated to SingletonHolder. Contains the actual storage buffer for the type T.
- struct SPSCQueue
- struct StaticMeshComponent
- struct StaticMeshPointDef
- struct StaticMeshResource
- struct StaticMeshTri
- struct StaticMeshVertex
- struct UIManager
- struct WeaponComponent
- struct World

Typedefs

- using EngineSingletons = SingletonHolder<InternalEngineSingletonsOrder>

Enumerations

- enum PawnState { PawnState_Ground , PawnState_Void }
- enum PawnStance { PawnStance_Stand , PawnStance_Crouch , PawnStance_Prone }
- enum EntityFlags : oxyU32 {
  EntityFlags_Disabled = 1 << 0 , EntityFlags_Static = 1 << 1 , EntityFlags_Dynamic = 1 << 2 ,
  EntityFlags_Renderable = 1 << 3 ,
  EntityFlags_HasHull = 1 << 4 , EntityFlags_HasCamera = 1 << 5 , EntityFlags_Replicated = 1 << 6 ,
  EntityFlags_IsLocalPlayer = 1 << 7 ,
  EntityFlags_EnableTransformReplication = 1 << 8 , EntityFlags_EnableTransformInterpolation = 1 << 9 }
- enum GfxRenderStrategy : oxyU8 { GfxRenderStrategy_DirectToGPU = 0 , GfxRenderStrategy_SoftwareDepthRasterizePreSo
  , GfxRenderStrategy_SoftwareDepthRasterizePreSortedOverlay , GfxRenderStrategy_SoftwareDepthRasterize
  }
- enum GfxCullType { GfxCullType_None = 0 , GfxCullType_Backface = 1 , GfxCullType_Frontface = 2 }
- enum ControllerButton : oxyU8 {
  ControllerButton_LeftThumb , ControllerButton_RightThumb , ControllerButton_LeftShoulder , ControllerButton_RightShoulder
  ,
  ControllerButton_South , ControllerButton_East , ControllerButton_West , ControllerButton_North ,
  ControllerButton_StartSelect , ControllerButton_BackShare , ControllerButton_DPadUp , ControllerButton_DPadDown
  ,
  ControllerButton_DPadLeft , ControllerButton_DPadRight , ControllerButton_Count }

- enum ControllerAxis : oxyU8 {
  ControllerAxis_LeftThumbX , ControllerAxis_LeftThumbY , ControllerAxis_RightThumbX , ControllerAxis_RightThumbY ,
  ControllerAxis_LeftTrigger , ControllerAxis_RightTrigger , ControllerAxis_Count }
- enum MouseButton : oxyU8 {
  MouseButton_Left , MouseButton_Right , MouseButton_Middle , MouseButton_X1 ,
  MouseButton_X2 , MouseButton_Count }
- enum KeyboardButton : oxyU8 {
  KeyboardButton_A , KeyboardButton_B , KeyboardButton_C , KeyboardButton_D ,
  KeyboardButton_E , KeyboardButton_F , KeyboardButton_G , KeyboardButton_H ,
  KeyboardButton_I , KeyboardButton_J , KeyboardButton_K , KeyboardButton_L ,
  KeyboardButton_M , KeyboardButton_N , KeyboardButton_O , KeyboardButton_P ,
  KeyboardButton_Q , KeyboardButton_R , KeyboardButton_S , KeyboardButton_T ,
  KeyboardButton_U , KeyboardButton_V , KeyboardButton_W , KeyboardButton_X ,
  KeyboardButton_Y , KeyboardButton_Z , KeyboardButton_0 , KeyboardButton_1 ,
  KeyboardButton_2 , KeyboardButton_3 , KeyboardButton_4 , KeyboardButton_5 ,
  KeyboardButton_6 , KeyboardButton_7 , KeyboardButton_8 , KeyboardButton_9 ,
  KeyboardButton_F1 , KeyboardButton_F2 , KeyboardButton_F3 , KeyboardButton_F4 ,
  KeyboardButton_F5 , KeyboardButton_F6 , KeyboardButton_F7 , KeyboardButton_F8 ,
  KeyboardButton_F9 , KeyboardButton_F10 , KeyboardButton_F11 , KeyboardButton_F12 ,
  KeyboardButton_F13 , KeyboardButton_F14 , KeyboardButton_F15 , KeyboardButton_F16 ,
  KeyboardButton_F17 , KeyboardButton_F18 , KeyboardButton_F19 , KeyboardButton_F20 ,
  KeyboardButton_F21 , KeyboardButton_F22 , KeyboardButton_F23 , KeyboardButton_F24 ,
  KeyboardButton_NumPad0 , KeyboardButton_NumPad1 , KeyboardButton_NumPad2 , KeyboardButton_NumPad3 ,
  KeyboardButton_NumPad4 , KeyboardButton_NumPad5 , KeyboardButton_NumPad6 , KeyboardButton_NumPad7 ,
  KeyboardButton_NumPad8 , KeyboardButton_NumPad9 , KeyboardButton_NumPadDecimal , KeyboardButton_NumPadEnter ,
  KeyboardButton_NumPadAdd , KeyboardButton_NumPadSubtract , KeyboardButton_NumPadMultiply ,
  KeyboardButton_NumPadDivide ,
  KeyboardButton_NumPadLock , KeyboardButton_Left , KeyboardButton_Right , KeyboardButton_Up ,
  KeyboardButton_Down , KeyboardButton_Home , KeyboardButton_End , KeyboardButton_PageUp ,
  KeyboardButton_PageDown , KeyboardButton_Insert , KeyboardButton_Delete , KeyboardButton_Pause ,
  KeyboardButton_PrintScreen , KeyboardButton_ScrollLock , KeyboardButton_Escape , KeyboardButton_Backtick ,
  KeyboardButton_Tab , KeyboardButton_CapsLock , KeyboardButton_LeftShift , KeyboardButton_LeftControl ,
  KeyboardButton_LeftWindows , KeyboardButton_LeftAlt , KeyboardButton_Space , KeyboardButton_RightAlt ,
  KeyboardButton_RightFunction , KeyboardButton_RightMenu , KeyboardButton_RightControl , KeyboardButton_RightShift ,
  KeyboardButton_Enter , KeyboardButton_Backspace , KeyboardButton_Comma , KeyboardButton_Period ,
  KeyboardButton_Slash , KeyboardButton_Semicolon , KeyboardButton_Apostrophe , KeyboardButton_LeftBracket ,
  KeyboardButton_RightBracket , KeyboardButton_Backslash , KeyboardButton_Hyphen , KeyboardButton_Equals ,
  KeyboardButton_Count }
- enum CollisionHull : oxyU8 {
  CollisionHull_None = 0xFF , CollisionHull_Point = 0 , CollisionHull_Player , CollisionHull_PlayerCrouched ,
  CollisionHull_Grenade }
- enum CollisionResponseType : oxyU8 { CollisionResponseType_None , CollisionResponseType_Slide ,
  CollisionResponseType_Bounce }
- enum EntitySummonType : oxyU8 { EntitySummonType_Player , EntitySummonType_Count }
- enum AnimationHash : oxyU32 {
  AnimationHash_Idle = 0x7c161a2b , AnimationHash_RunForward = 0x947ec374 , AnimationHash_RunBackward
  = 0x144ff8d , AnimationHash_Dying = 0x12c8a4ff ,
  AnimationHash_Throw = 0x8e526e33 }

- enum EntitySpawnType : oxyU8 {
  EntitySpawnType_Player , EntitySpawnType_Golfclub , EntitySpawnType_GolfclubLauncher , EntitySpawnType_Golfball
  ,
  EntitySpawntype_GolfballLauncher , EntitySpawnType_Count }
- enum NetProtoMsgType : oxyU16 {
  NetProtoMsgType_AnyPing = 0 , NetProtoMsgType_SrvWelcome , NetProtoMsgType_SrvChangeLevel ,
  NetProtoMsgType_SrvSetLocalPlayer ,
  NetProtoMsgType_SrvEntitySpawn , NetProtoMsgType_SrvEntityDestroy , NetProtoMsgType_SrvEntityTransformRepl
  , NetProtoMsgType_SrvHealhComponentChange ,
  NetProtoMsgType_SrvPawnPickupWeapon , NetProtoMsgType_SrvPawnDropWeapon , NetProtoMsgType_CliLocalPlayerEntit
  , NetProtoMsgType_CliPawnDropWeapon ,
  NetProtoMsgType_CliLocalPlayerFireWeapon }
- enum PickupType : oxyU8 { PickupType_Health , PickupType_Ammo , PickupType_Weapon ,
  PickupType_Count }
- enum WeaponFireType : oxyU8 { WeaponFireType_Bullets , WeaponFireType_GolfClub , WeaponFireType_GolfBall
  , WeaponFireType_Count }
- enum HealthState : oxyU8 { HealthState_Alive , HealthState_Invulnerable , HealthState_Dead }
- enum DamageType : oxyU8 {
  DamageType_None , DamageType_Explosive , DamageType_Bullet , DamageType_Melee ,
  DamageType_FallDamage , DamageType_Count }

Functions

- auto CullBackfaceTri (const GfxTri &tri) -> bool
- auto CullFrontfaceTri (const GfxTri &tri) -> bool
- constexpr auto CRC64Eval (const oxyU8 *data, oxySize size) -> oxyU64
- auto RandomS32 (oxyS32 minInclusive, oxyS32 maxInclusive) -> oxyS32
- auto RandomU32 (oxyU32 minInclusive, oxyU32 maxInclusive) -> oxyU32
- auto RandomS64 (oxyS64 minInclusive, oxyS64 maxInclusive) -> oxyS64
- auto RandomU64 (oxyU64 minInclusive, oxyU64 maxInclusive) -> oxyU64
- auto RandomF32 (oxyF32 minInclusive, oxyF32 maxInclusive) -> oxyF32
- auto RandomF64 (oxyF64 minInclusive, oxyF64 maxInclusive) -> oxyF64
- auto RandomBool () -> oxyBool
- auto GetObjectDescriptionMap () -> std::unordered_map< oxyU64, const ObjectDescription * > &
- auto GetExecutableDirectory () -> std::string_view
- auto GetLaunchArguments () -> std::span< const std::string >
- auto LogMessage (const char *str) -> void
- auto ReadFileContents (std::string_view absolutePath) -> std::vector< oxyU8 >
- auto CreateFileMap (std::string_view path, oxyBool write, oxySize requestSize) -> UniqueFileMap
- auto Win64PlatformInit () -> void
- auto Win64PlatformRender () -> void
- auto Win64PlatformUpdate (float deltaTimeSeconds) -> void
- auto Win64PlatformShutdown () -> void
- auto LoadWorld (std::string_view name) -> std::shared_ptr< World >

Variables

- constexpr oxyU64 g_CRC64Table [ ]
- std::random_device g_randomDevice
- std::mt19937_64 g_randomEngine {g_randomDevice()}
- constexpr oxyVec3 k_collisionHullMins [ ]
- constexpr oxyVec3 k_collisionHullMaxs [ ]

5.1.1 Typedef Documentation

5.1.1.1 EngineSingletons

```
using oxygen::EngineSingletons = SingletonHolder<InternalEngineSingletonsOrder>
```

5.1.2 Enumeration Type Documentation

5.1.2.1 AnimationHash

```
enum oxygen::AnimationHash : oxyU32
```

Enumerator

AnimationHash_Idle	
AnimationHash_RunForward	
AnimationHash_RunBackward	
AnimationHash_Dying	
AnimationHash_Throw	

5.1.2.2 CollisionHull

```
enum oxygen::CollisionHull : oxyU8
```

Enumerator

CollisionHull_None	
CollisionHull_Point	
CollisionHull_Player	
CollisionHull_PlayerCrouched	
CollisionHull_Grenade	

5.1.2.3 CollisionResponseType

```
enum oxygen::CollisionResponseType : oxyU8
```

Enumerator

CollisionResponseType_None	
CollisionResponseType_Slide	
CollisionResponseType_Bounce	

5.1.2.4 ControllerAxis

```
enum oxygen::ControllerAxis : oxyU8
```

Enumerator

ControllerAxis_LeftThumbX	
ControllerAxis_LeftThumbY	
ControllerAxis_RightThumbX	
ControllerAxis_RightThumbY	
ControllerAxis_LeftTrigger	
ControllerAxis_RightTrigger	
ControllerAxis_Count	

5.1.2.5 ControllerButton

`enum oxygen::ControllerButton : oxyU8`

Enumerator

ControllerButton_LeftThumb	
ControllerButton_RightThumb	
ControllerButton_LeftShoulder	
ControllerButton_RightShoulder	
ControllerButton_South	
ControllerButton_East	
ControllerButton_West	
ControllerButton_North	
ControllerButton_StartSelect	
ControllerButton_BackShare	
ControllerButton_DPadUp	
ControllerButton_DPadDown	
ControllerButton_DPadLeft	
ControllerButton_DPadRight	
ControllerButton_Count	

5.1.2.6 DamageType

`enum oxygen::DamageType : oxyU8`

Enumerator

DamageType_None	
DamageType_Explosive	
DamageType_Bullet	
DamageType_Melee	
DamageType_FallDamage	
DamageType_Count	

5.1.2.7 EntityFlags

`enum oxygen::EntityFlags : oxyU32`

Enumerator

EntityFlags_Disabled	
EntityFlags_Static	
EntityFlags_Dynamic	
EntityFlags_Renderable	
EntityFlags_HasHull	
EntityFlags_HasCamera	
EntityFlags_Replicated	
EntityFlags_IsLocalPlayer	
EntityFlags_EnableTransformReplication	
EntityFlags_EnableTransformInterpolation	

5.1.2.8 EntitySpawnType

`enum oxygen::EntitySpawnType : oxyU8`

Enumerator

EntitySpawnType_Player	
EntitySpawnType_Golfclub	
EntitySpawnType_GolfclubLauncher	
EntitySpawnType_Golfball	
EntitySpawntype_GolfballLauncher	
EntitySpawnType_Count	

5.1.2.9 EntitySummonType

`enum oxygen::EntitySummonType : oxyU8`

Enumerator

EntitySummonType_Player	
EntitySummonType_Count	

5.1.2.10 GfxCullType

`enum oxygen::GfxCullType`

Enumerator

GfxCullType_None	
GfxCullType_Backface	
GfxCullType_Frontface	

5.1.2.11 GfxRenderStrategy

`enum oxygen::GfxRenderStrategy : oxyU8`

Enumerator

GfxRenderStrategy_DirectToGPU	
GfxRenderStrategy_SoftwareDepthRasterizePreSorted	
GfxRenderStrategy_SoftwareDepthRasterizePreSortedOverlay	
GfxRenderStrategy_SoftwareDepthRasterize	

5.1.2.12 HealthState

`enum oxygen::HealthState : oxyU8`

Enumerator

HealthState_Alive	
HealthState_Invulnerable	
HealthState_Dead	

5.1.2.13 KeyboardButton

`enum oxygen::KeyboardButton : oxyU8`

Enumerator

KeyboardButton_A	
KeyboardButton_B	
KeyboardButton_C	
KeyboardButton_D	
KeyboardButton_E	
KeyboardButton_F	
KeyboardButton_G	
KeyboardButton_H	
KeyboardButton_I	
KeyboardButton_J	
KeyboardButton_K	
KeyboardButton_L	
KeyboardButton_M	
KeyboardButton_N	
KeyboardButton_O	
KeyboardButton_P	
KeyboardButton_Q	
KeyboardButton_R	
KeyboardButton_S	
KeyboardButton_T	
KeyboardButton_U	
KeyboardButton_V	
KeyboardButton_W	
KeyboardButton_X	

Enumerator

KeyboardButton_Y	
KeyboardButton_Z	
KeyboardButton_0	
KeyboardButton_1	
KeyboardButton_2	
KeyboardButton_3	
KeyboardButton_4	
KeyboardButton_5	
KeyboardButton_6	
KeyboardButton_7	
KeyboardButton_8	
KeyboardButton_9	
KeyboardButton_F1	
KeyboardButton_F2	
KeyboardButton_F3	
KeyboardButton_F4	
KeyboardButton_F5	
KeyboardButton_F6	
KeyboardButton_F7	
KeyboardButton_F8	
KeyboardButton_F9	
KeyboardButton_F10	
KeyboardButton_F11	
KeyboardButton_F12	
KeyboardButton_F13	
KeyboardButton_F14	
KeyboardButton_F15	
KeyboardButton_F16	
KeyboardButton_F17	
KeyboardButton_F18	
KeyboardButton_F19	
KeyboardButton_F20	
KeyboardButton_F21	
KeyboardButton_F22	
KeyboardButton_F23	
KeyboardButton_F24	
KeyboardButton_NumPad0	
KeyboardButton_NumPad1	
KeyboardButton_NumPad2	
KeyboardButton_NumPad3	
KeyboardButton_NumPad4	
KeyboardButton_NumPad5	
KeyboardButton_NumPad6	
KeyboardButton_NumPad7	
KeyboardButton_NumPad8	
KeyboardButton_NumPad9	
KeyboardButton_NumPadDecimal	

Enumerator

KeyboardButton_NumPadEnter	
KeyboardButton_NumPadAdd	
KeyboardButton_NumPadSubtract	
KeyboardButton_NumPadMultiply	
KeyboardButton_NumPadDivide	
KeyboardButton_NumPadLock	
KeyboardButton_Left	
KeyboardButton_Right	
KeyboardButton_Up	
KeyboardButton_Down	
KeyboardButton_Home	
KeyboardButton_End	
KeyboardButton_PageUp	
KeyboardButton_PageDown	
KeyboardButton_Insert	
KeyboardButton_Delete	
KeyboardButton_Pause	
KeyboardButton_PrintScreen	
KeyboardButton_ScrollLock	
KeyboardButton_Escape	
KeyboardButton_Backtick	
KeyboardButton_Tab	
KeyboardButton_CapsLock	
KeyboardButton_LeftShift	
KeyboardButton_LeftControl	
KeyboardButton_LeftWindows	
KeyboardButton_LeftAlt	
KeyboardButton_Space	
KeyboardButton_RightAlt	
KeyboardButton_RightFunction	
KeyboardButton_RightMenu	
KeyboardButton_RightControl	
KeyboardButton_RightShift	
KeyboardButton_Enter	
KeyboardButton_Backspace	
KeyboardButton_Comma	
KeyboardButton_Period	
KeyboardButton_Slash	
KeyboardButton_Semicolon	
KeyboardButton_Apostrophe	
KeyboardButton_LeftBracket	
KeyboardButton_RightBracket	
KeyboardButton_Backslash	
KeyboardButton_Hyphen	
KeyboardButton_Equals	
KeyboardButton_Count	

5.1.2.14 MouseButton

enum oxygen::MouseButton : oxyU8

Enumerator

MouseButton_Left	
MouseButton_Right	
MouseButton_Middle	
MouseButton_X1	
MouseButton_X2	
MouseButton_Count	

5.1.2.15 NetProtoMsgType

enum oxygen::NetProtoMsgType : oxyU16

Enumerator

NetProtoMsgType_AnyPing	
NetProtoMsgType_SrvWelcome	
NetProtoMsgType_SrvChangeLevel	
NetProtoMsgType_SrvSetLocalPlayer	
NetProtoMsgType_SrvEntitySpawn	
NetProtoMsgType_SrvEntityDestroy	
NetProtoMsgType_SrvEntityTransformRepl	
NetProtoMsgType_SrvHealhComponentChange	
NetProtoMsgType_SrvPawnPickupWeapon	
NetProtoMsgType_SrvPawnDropWeapon	
NetProtoMsgType_CliLocalPlayerEntityMove	
NetProtoMsgType_CliPawnDropWeapon	
NetProtoMsgType_CliLocalPlayerFireWeapon	

5.1.2.16 PawnStance

enum oxygen::PawnStance

Enumerator

PawnStance_Stand	
PawnStance_Crouch	
PawnStance_Prone	

5.1.2.17 PawnState

enum oxygen::PawnState

Enumerator

PawnState_Ground	
PawnState_Void	

5.1.2.18 PickupType

```
enum oxygen::PickupType :  oxyU8
```

Enumerator

PickupType_Health	
PickupType_Ammo	
PickupType_Weapon	
PickupType_Count	

5.1.2.19 WeaponFireType

```
enum oxygen::WeaponFireType :  oxyU8
```

Enumerator

WeaponFireType_Bullets	
WeaponFireType_GolfClub	
WeaponFireType_GolfBall	
WeaponFireType_Count	

5.1.3 Function Documentation

5.1.3.1 CRC64Eval()

```
auto oxygen::CRC64Eval (
            const oxyU8 * data,
            oxySize size) -> oxyU64   [inline], [constexpr]
```

5.1.3.2 CreateFileMap()

```
auto oxygen::CreateFileMap (
            std::string_view path,
            oxyBool write,
            oxySize requestSize) -> UniqueFileMap
```

5.1.3.3 CullBackfaceTri()

```
auto oxygen::CullBackfaceTri (
            const GfxTri & tri) -> bool   [inline]
```

5.1.3.4 CullFrontfaceTri()

```
auto oxygen::CullFrontfaceTri (
            const GfxTri & tri) -> bool   [inline]
```

5.1.3.5 GetExecutableDirectory()

```
auto oxygen::GetExecutableDirectory () -> std::string_view
```

5.1.3.6 GetLaunchArguments()

```
auto oxygen::GetLaunchArguments () -> std::span<const std::string>
```

5.1.3.7 GetObjectDescriptionMap()

```
auto oxygen::GetObjectDescriptionMap () -> std::unordered_map<oxyU64, const ObjectDescription*>&
[inline]
```

5.1.3.8 LoadWorld()

```
auto oxygen::LoadWorld (
            std::string_view name) -> std::shared_ptr<World>
```

5.1.3.9 LogMessage()

```
auto oxygen::LogMessage (
            const char * str) -> void
```

5.1.3.10 RandomBool()

```
auto oxygen::RandomBool () -> oxyBool   [inline]
```

5.1.3.11 RandomF32()

```
auto oxygen::RandomF32 (
            oxyF32 minInclusive,
            oxyF32 maxInclusive) -> oxyF32   [inline]
```

5.1.3.12 RandomF64()

```
auto oxygen::RandomF64 (
            oxyF64 minInclusive,
            oxyF64 maxInclusive) -> oxyF64   [inline]
```

5.1.3.13 RandomS32()

```
auto oxygen::RandomS32 (
            oxyS32 minInclusive,
            oxyS32 maxInclusive) -> oxyS32   [inline]
```

5.1.3.14 RandomS64()

```
auto oxygen::RandomS64 (
            oxyS64 minInclusive,
            oxyS64 maxInclusive) -> oxyS64   [inline]
```

5.1.3.15 RandomU32()

```
auto oxygen::RandomU32 (
            oxyU32 minInclusive,
            oxyU32 maxInclusive) -> oxyU32   [inline]
```

5.1.3.16 RandomU64()

```
auto oxygen::RandomU64 (
            oxyU64 minInclusive,
            oxyU64 maxInclusive) -> oxyU64   [inline]
```

5.1.3.17 ReadFileContents()

```
auto oxygen::ReadFileContents (
            std::string_view absolutePath) -> std::vector<oxyU8>
```

5.1.3.18 Win64PlatformInit()

```
auto oxygen::Win64PlatformInit () -> void
```

5.1.3.19 Win64PlatformRender()

```
auto oxygen::Win64PlatformRender () -> void
```

5.1.3.20 Win64PlatformShutdown()

```
auto oxygen::Win64PlatformShutdown () -> void
```

5.1.3.21 Win64PlatformUpdate()

```
auto oxygen::Win64PlatformUpdate (
            float deltaTimeSeconds) -> void
```

5.1.4 Variable Documentation

5.1.4.1 g_CRC64Table

```
oxyU64 oxygen::g_CRC64Table[]  [inline], [constexpr]
```

5.1.4.2 g_randomDevice

```
std::random_device oxygen::g_randomDevice  [inline]
```

5.1.4.3 g_randomEngine

```
std::mt19937_64 oxygen::g_randomEngine {g_randomDevice()}  [inline]
```

5.1.4.4 k_collisionHullMaxs

```
oxyVec3 oxygen::k_collisionHullMaxs[]  [inline], [constexpr]
```

Initial value:

```
= {
        oxyVec3{0.f, 0.f, 0.f},
        oxyVec3{24.f, 24.f, 48.f},
        oxyVec3{24.f, 24.f, 48.f},
        oxyVec3{12.f, 12.f, 12.f},
    }
```

5.1.4.5 k_collisionHullMins

```
oxyVec3 oxygen::k_collisionHullMins[]  [inline], [constexpr]
```

Initial value:

```
= {
        oxyVec3{0.f, 0.f, 0.f},
        oxyVec3{-24.f, -24.f, -48.f},
        oxyVec3{-24.f, -24.f, -48.f},
        oxyVec3{-12.f, -12.f, -12.f},
    }
```

5.2 oxygen::AudioAbstraction Namespace Reference

5.3 oxygen::BSPDefines Namespace Reference

Classes

- struct ClipNode
- struct Edge
- struct Face
- struct Header
- struct Leaf
- struct Lump
- struct MipTex
- struct MipTexLump
- struct Model
- struct Node
- struct Plane
- struct TexInfo
- struct Vertex

Enumerations

- enum Contents {
  Contents_Empty = -1 , Contents_Solid = -2 , Contents_Water = -3 , Contents_Slime = -4 ,
  Contents_Lava = -5 , Contents_Sky = -6 , Contents_Origin = -7 , Contents_Clip = -8 ,
  Contents_Current0 = -9 , Contents_Current90 = -10 , Contents_Current180 = -11 , Contents_Current270 = -12 ,
  Contents_CurrentUp = -13 , Contents_CurrentDown = -14 , Contents_Translucent = -15 }
- enum LumpIndex {
  LumpIndex_Entities = 0 , LumpIndex_Planes , LumpIndex_Textures , LumpIndex_Vertexes ,
  LumpIndex_Visibility , LumpIndex_Nodes , LumpIndex_TexInfo , LumpIndex_Faces ,
  LumpIndex_Lighting , LumpIndex_ClipNodes , LumpIndex_Leafs , LumpIndex_MarkSurfaces ,
  LumpIndex_Edges , LumpIndex_SurfEdges , LumpIndex_Models , LumpIndex_Count }
- enum PlaneType {
  Plane_X = 0 , Plane_Y , Plane_Z , Plane_AnyX ,
  Plane_AnyY , Plane_AnyZ }

Variables

- constexpr auto k_BSPVersion = oxyS32{30}
- constexpr auto k_ToolVersion = oxyS32{2}
- constexpr auto k_MaxMapHulls = oxySize{4}
- constexpr auto k_MaxMapModels = oxySize{400}
- constexpr auto k_MaxMapBrushes = oxySize{4096}
- constexpr auto k_MaxMapEntityString = oxySize{128 ∗ 1024}
- constexpr auto k_MaxMapPlanes = oxySize{32767}
- constexpr auto k_MaxMapNodes = oxySize{32767}
- constexpr auto k_MaxMapClipNodes = oxySize{32767}
- constexpr auto k_MaxMapLeafs = oxySize{8192}
- constexpr auto k_MaxMapVertices = oxySize{65535}
- constexpr auto k_MaxMapFaces = oxySize{65535}
- constexpr auto k_MaxMapMarkSurfaces = oxySize{65535}
- constexpr auto k_MaxMapTexInfo = oxySize{8192}
- constexpr auto k_MaxMapEdges = oxySize{256000}
- constexpr auto k_MaxMapSurfEdges = oxySize{512000}
- constexpr auto k_MaxMapTextures = oxySize{512}
- constexpr auto k_MaxMapMipTex = oxySize{0x200000}
- constexpr auto k_MaxMapLighting = oxySize{0x200000}
- constexpr auto k_MaxMapVis = oxySize{0x200000}
- constexpr auto k_MaxMapPortals = oxySize{65536}
- constexpr auto k_NumMipLevels = 4
- constexpr auto k_TexSpecial
- constexpr auto k_MaxLightMaps = 4
- constexpr auto k_NumAmbients = 4

5.3.1 Enumeration Type Documentation

5.3.1.1 Contents

enum oxygen::BSPDefines::Contents

Enumerator

Contents_Empty	
Contents_Solid	
Contents_Water	
Contents_Slime	
Contents_Lava	
Contents_Sky	
Contents_Origin	
Contents_Clip	
Contents_Current0	
Contents_Current90	
Contents_Current180	
Contents_Current270	
Contents_CurrentUp	
Contents_CurrentDown	
Contents_Translucent	

5.3.1.2 LumpIndex

`enum oxygen::BSPDefines::LumpIndex`

Enumerator

LumpIndex_Entities	
LumpIndex_Planes	
LumpIndex_Textures	
LumpIndex_Vertexes	
LumpIndex_Visibility	
LumpIndex_Nodes	
LumpIndex_TexInfo	
LumpIndex_Faces	
LumpIndex_Lighting	
LumpIndex_ClipNodes	
LumpIndex_Leafs	
LumpIndex_MarkSurfaces	
LumpIndex_Edges	
LumpIndex_SurfEdges	
LumpIndex_Models	
LumpIndex_Count	

5.3.1.3 PlaneType

`enum oxygen::BSPDefines::PlaneType`

Enumerator

Plane_X	
Plane_Y	
Plane_Z	
Plane_AnyX	
Plane_AnyY	
Plane_AnyZ	

5.3.2 Variable Documentation

5.3.2.1 k_BSPVersion

```
auto oxygen::BSPDefines::k_BSPVersion = oxyS32{30}  [inline], [constexpr]
```

5.3.2.2 k_MaxLightMaps

```
auto oxygen::BSPDefines::k_MaxLightMaps = 4  [inline], [constexpr]
```

5.3.2.3 k_MaxMapBrushes

```
auto oxygen::BSPDefines::k_MaxMapBrushes = oxySize{4096}  [inline], [constexpr]
```

5.3.2.4 k_MaxMapClipNodes

```
auto oxygen::BSPDefines::k_MaxMapClipNodes = oxySize{32767}  [inline], [constexpr]
```

5.3.2.5 k_MaxMapEdges

```
auto oxygen::BSPDefines::k_MaxMapEdges = oxySize{256000}  [inline], [constexpr]
```

5.3.2.6 k_MaxMapEntityString

```
auto oxygen::BSPDefines::k_MaxMapEntityString = oxySize{128 * 1024}  [inline], [constexpr]
```

5.3.2.7 k_MaxMapFaces

```
auto oxygen::BSPDefines::k_MaxMapFaces = oxySize{65535}  [inline], [constexpr]
```

5.3.2.8 k_MaxMapHulls

```
auto oxygen::BSPDefines::k_MaxMapHulls = oxySize{4}  [inline], [constexpr]
```

5.3.2.9 k_MaxMapLeafs

```
auto oxygen::BSPDefines::k_MaxMapLeafs = oxySize{8192}  [inline], [constexpr]
```

5.3.2.10 k_MaxMapLighting

```
auto oxygen::BSPDefines::k_MaxMapLighting = oxySize{0x200000}  [inline], [constexpr]
```

5.3.2.11 k_MaxMapMarkSurfaces

```
auto oxygen::BSPDefines::k_MaxMapMarkSurfaces = oxySize{65535}  [inline], [constexpr]
```

5.3.2.12 k_MaxMapMipTex

```
auto oxygen::BSPDefines::k_MaxMapMipTex = oxySize{0x200000}  [inline], [constexpr]
```

5.3.2.13 k_MaxMapModels

```
auto oxygen::BSPDefines::k_MaxMapModels = oxySize{400}  [inline], [constexpr]
```

5.3.2.14 k_MaxMapNodes

```
auto oxygen::BSPDefines::k_MaxMapNodes = oxySize{32767}  [inline], [constexpr]
```

5.3.2.15 k_MaxMapPlanes

```
auto oxygen::BSPDefines::k_MaxMapPlanes = oxySize{32767}  [inline], [constexpr]
```

5.3.2.16 k_MaxMapPortals

```
auto oxygen::BSPDefines::k_MaxMapPortals = oxySize{65536}  [inline], [constexpr]
```

5.3.2.17 k_MaxMapSurfEdges

```
auto oxygen::BSPDefines::k_MaxMapSurfEdges = oxySize{512000}  [inline], [constexpr]
```

5.3.2.18 k_MaxMapTexInfo

```
auto oxygen::BSPDefines::k_MaxMapTexInfo = oxySize{8192}  [inline], [constexpr]
```

5.3.2.19 k_MaxMapTextures

```
auto oxygen::BSPDefines::k_MaxMapTextures = oxySize{512}  [inline], [constexpr]
```

5.3.2.20 k_MaxMapVertices

```
auto oxygen::BSPDefines::k_MaxMapVertices = oxySize{65535}  [inline], [constexpr]
```

5.3.2.21 k_MaxMapVis

```
auto oxygen::BSPDefines::k_MaxMapVis = oxySize{0x200000}  [inline], [constexpr]
```

5.3.2.22 k_NumAmbients

```
auto oxygen::BSPDefines::k_NumAmbients = 4  [inline], [constexpr]
```

5.3.2.23 k_NumMipLevels

```
auto oxygen::BSPDefines::k_NumMipLevels = 4  [inline], [constexpr]
```

5.3.2.24 k_TexSpecial

```
auto oxygen::BSPDefines::k_TexSpecial  [inline], [constexpr]
```

Initial value:

```
=
          1
```

5.3.2.25 k_ToolVersion

```
auto oxygen::BSPDefines::k_ToolVersion = oxyS32{2}  [inline], [constexpr]
```

5.4 oxygen::GfxSoftwareRasterizer Namespace Reference

Classes

- struct CountingIterator

Functions

- auto RasterTriDepthTest (const GfxTri &tri, oxyS16 triID, oxyU32 width, oxyU32 height, oxyF32 *zbuffer, oxyS16 *tribuffer, oxyU32 divminx, oxyU32 divminy, oxyU32 divmaxx, oxyU32 divmaxy) -> oxyBool
- auto RasterTriNoDepthCompare (const GfxTri &tri, oxyU32 width, oxyU32 height, oxyF32 *zbuffer, oxyU32 divminx, oxyU32 divminy, oxyU32 divmaxx, oxyU32 divmaxy) -> void

5.4.1 Function Documentation

5.4.1.1 RasterTriDepthTest()

```
auto oxygen::GfxSoftwareRasterizer::RasterTriDepthTest (
            const GfxTri & tri,
            oxyS16 triID,
            oxyU32 width,
            oxyU32 height,
            oxyF32 * zbuffer,
            oxyS16 * tribuffer,
            oxyU32 divminx,
            oxyU32 divminy,
            oxyU32 divmaxx,
            oxyU32 divmaxy) -> oxyBool   [inline]
```

5.4.1.2 RasterTriNoDepthCompare()

```
auto oxygen::GfxSoftwareRasterizer::RasterTriNoDepthCompare (
            const GfxTri & tri,
            oxyU32 width,
            oxyU32 height,
            oxyF32 * zbuffer,
            oxyU32 divminx,
            oxyU32 divminy,
            oxyU32 divmaxx,
            oxyU32 divmaxy) -> void   [inline]
```

5.5 oxygen::GraphicsAbstraction Namespace Reference

Functions

- auto GetWindowSize (oxyS32 &width, oxyS32 &height) -> void
- auto LoadTexture (const char *absolutePath) -> std::shared_ptr< const Texture >
- auto DrawTexturedQuad (const TexturedQuad &quad) -> void

5.5.1 Function Documentation

5.5.1.1 DrawTexturedQuad()

```
auto oxygen::GraphicsAbstraction::DrawTexturedQuad (
            const TexturedQuad & quad) -> void
```

5.5.1.2 GetWindowSize()

```
auto oxygen::GraphicsAbstraction::GetWindowSize (
            oxyS32 & width,
            oxyS32 & height) -> void
```

5.5.1.3 LoadTexture()

```
auto oxygen::GraphicsAbstraction::LoadTexture (
            const char * absolutePath) -> std::shared_ptr<const Texture>
```

5.6 oxygen::InputAbstraction Namespace Reference

Functions

- auto IsForeground () -> oxyBool
- auto HideAndLockCursor (oxyBool lock) -> void
- auto GetMousePosition (oxyF32 &x, oxyF32 &y) -> void
- auto GetMouseStates (std::bitset< MouseButton_Count > &buttons) -> void
- auto GetKeyStates (std::bitset< KeyboardButton_Count > &keys) -> void
- auto GetControllerConnected (int index) -> oxyBool
- auto GetControllerStates (int index, std::bitset< ControllerButton_Count > &buttons) -> void
- auto GetControllerAxisStates (int index, std::span< oxyF32, ControllerAxis_Count > axes) -> void

5.6.1 Function Documentation

5.6.1.1 GetControllerAxisStates()

```
auto oxygen::InputAbstraction::GetControllerAxisStates (
            int index,
            std::span< oxyF32, ControllerAxis_Count > axes) -> void
```

5.6.1.2 GetControllerConnected()

```
auto oxygen::InputAbstraction::GetControllerConnected (
            int index) -> oxyBool
```

5.6.1.3 GetControllerStates()

```
auto oxygen::InputAbstraction::GetControllerStates (
            int index,
            std::bitset< ControllerButton_Count > & buttons) -> void
```

5.6.1.4 GetKeyStates()

```
auto oxygen::InputAbstraction::GetKeyStates (
            std::bitset< KeyboardButton_Count > & keys) -> void
```

5.6.1.5 GetMousePosition()

```
auto oxygen::InputAbstraction::GetMousePosition (
            oxyF32 & x,
            oxyF32 & y) -> void
```

5.6.1.6 GetMouseStates()

```
auto oxygen::InputAbstraction::GetMouseStates (
            std::bitset< MouseButton_Count > & buttons) -> void
```

5.6.1.7 HideAndLockCursor()

```
auto oxygen::InputAbstraction::HideAndLockCursor (
            oxyBool lock) -> void
```

5.6.1.8 IsForeground()

```
auto oxygen::InputAbstraction::IsForeground () -> oxyBool
```

5.7 oxygen::Math Namespace Reference

Functions

- constexpr auto Translate (const oxyMat4x4 &m, const oxyVec3 &v) -> oxyMat4x4
- constexpr auto Rotate (const oxyMat4x4 &m, const oxyQuat &q) -> oxyMat4x4
- auto Rotate (const oxyMat4x4 &m, oxyF32 angle, const oxyVec3 &axis) -> oxyMat4x4
- constexpr auto Scale (const oxyMat4x4 &m, const oxyVec3 &v) -> oxyMat4x4
- auto LookAt (const oxyVec3 &eye, const oxyVec3 &center, const oxyVec3 &up) -> oxyMat4x4
- auto Perspective (oxyF32 fovy, oxyF32 aspect, oxyF32 near, oxyF32 far) -> oxyMat4x4
- auto InverseMatrix (const oxyMat4x4 &m) -> oxyMat4x4
- auto RotationMatrixToEuler (const oxyMat4x4 &m) -> oxyVec3
- auto Slerp (const oxyQuat &a, const oxyQuat &b, oxyF32 t) -> oxyQuat
- auto AngleAxisToQuat (const oxyF32 angle, const oxyVec3 &axis) -> oxyQuat
- auto QuatToEulerAngles (const oxyQuat &q) -> oxyVec3
- auto EulerAnglesToQuat (const oxyVec3 &v) -> oxyQuat
- auto QuatLookAt (const oxyVec3 &position, const oxyVec3 &where) -> oxyQuat
- auto EulerForward (const oxyVec3 &euler) -> oxyVec3
- auto ToHalfFloat (oxyF32 x) -> oxyU16
- auto FromHalfFloat (oxyU16 x) -> oxyF32

5.7.1 Function Documentation

5.7.1.1 AngleAxisToQuat()

```
auto oxygen::Math::AngleAxisToQuat (
            const oxyF32 angle,
            const oxyVec3 & axis) -> oxyQuat   [inline]
```

5.7.1.2 EulerAnglesToQuat()

```
auto oxygen::Math::EulerAnglesToQuat (
            const oxyVec3 & v) -> oxyQuat   [inline]
```

5.7.1.3 EulerForward()

```
auto oxygen::Math::EulerForward (
            const oxyVec3 & euler) -> oxyVec3   [inline]
```

5.7.1.4 FromHalfFloat()

```
auto oxygen::Math::FromHalfFloat (
            oxyU16 x) -> oxyF32   [inline]
```

5.7.1.5 InverseMatrix()

```
auto oxygen::Math::InverseMatrix (
            const oxyMat4x4 & m) -> oxyMat4x4   [inline]
```

5.7.1.6 LookAt()

```
auto oxygen::Math::LookAt (
            const oxyVec3 & eye,
            const oxyVec3 & center,
            const oxyVec3 & up) -> oxyMat4x4   [inline]
```

5.7.1.7 Perspective()

```
auto oxygen::Math::Perspective (
            oxyF32 fovy,
            oxyF32 aspect,
            oxyF32 near,
            oxyF32 far) -> oxyMat4x4   [inline]
```

5.7.1.8 QuatLookAt()

```
auto oxygen::Math::QuatLookAt (
            const oxyVec3 & position,
            const oxyVec3 & where) -> oxyQuat   [inline]
```

5.7.1.9 QuatToEulerAngles()

```
auto oxygen::Math::QuatToEulerAngles (
            const oxyQuat & q) -> oxyVec3   [inline]
```

5.7.1.10 Rotate() [1/2]

```
auto oxygen::Math::Rotate (
            const oxyMat4x4 & m,
            const oxyQuat & q) -> oxyMat4x4   [inline], [constexpr]
```

5.7.1.11 Rotate() [2/2]

```
auto oxygen::Math::Rotate (
            const oxyMat4x4 & m,
            oxyF32 angle,
            const oxyVec3 & axis) -> oxyMat4x4   [inline]
```

5.7.1.12 RotationMatrixToEuler()

```
auto oxygen::Math::RotationMatrixToEuler (
            const oxyMat4x4 & m) -> oxyVec3   [inline]
```

5.7.1.13 Scale()

```
auto oxygen::Math::Scale (
            const oxyMat4x4 & m,
            const oxyVec3 & v) -> oxyMat4x4   [inline], [constexpr]
```

5.7.1.14 Slerp()

```
auto oxygen::Math::Slerp (
            const oxyQuat & a,
            const oxyQuat & b,
            oxyF32 t) -> oxyQuat   [inline]
```

5.7.1.15 ToHalfFloat()

```
auto oxygen::Math::ToHalfFloat (
            oxyF32 x) -> oxyU16   [inline]
```

5.7.1.16 Translate()

```
auto oxygen::Math::Translate (
            const oxyMat4x4 & m,
            const oxyVec3 & v) -> oxyMat4x4   [inline], [constexpr]
```

5.8 oxygen::NetworkAbstraction Namespace Reference

Functions

- auto ConnectToHost (const char *host, oxyU16 port) -> std::unique_ptr< NetworkSocket >
- auto HostServer (oxyU16 port) -> std::unique_ptr< NetworkSocket >
- auto CreateBroadcastSendSocket (oxyU16 port) -> std::unique_ptr< NetworkSocket >
- auto CreateBroadcastListenSocket (oxyU16 port) -> std::unique_ptr< NetworkSocket >

5.8.1 Function Documentation

5.8.1.1 ConnectToHost()

```
auto oxygen::NetworkAbstraction::ConnectToHost (
            const char * host,
            oxyU16 port) -> std::unique_ptr<NetworkSocket>
```

5.8.1.2 CreateBroadcastListenSocket()

```
auto oxygen::NetworkAbstraction::CreateBroadcastListenSocket (
            oxyU16 port) -> std::unique_ptr<NetworkSocket>
```

5.8.1.3 CreateBroadcastSendSocket()

```
auto oxygen::NetworkAbstraction::CreateBroadcastSendSocket (
            oxyU16 port) -> std::unique_ptr<NetworkSocket>
```

5.8.1.4 HostServer()

```
auto oxygen::NetworkAbstraction::HostServer (
            oxyU16 port) -> std::unique_ptr<NetworkSocket>
```

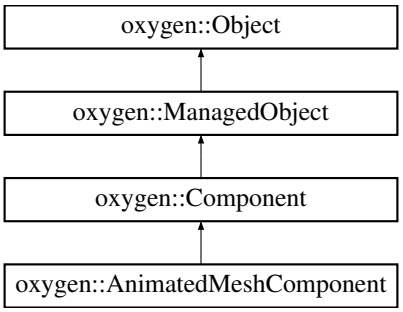
Chapter 6

Class Documentation

6.1 oxygen::AnimatedMeshComponent Struct Reference

`#include <AnimatedMeshComponent.h>`

Inheritance diagram for oxygen::AnimatedMeshComponent:



Public Member Functions

- OXYGENOBJECT (AnimatedMeshComponent, Component)
- auto LoadByName (std::string_view name) -> oxyBool
- auto BeginAnimation (oxyU32 animHash, oxyBool loop=true) -> void
- auto GetCurrenntAnimationHash () const -> oxyU32
- auto SetLocalOffset (const oxyVec3 &offset) -> void
- auto SetLocalRotation (const oxyQuat &rotation) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (float deltaTimeSeconds) -> void override
- auto Render () const -> void override

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.1.1 Member Function Documentation

6.1.1.1 BeginAnimation()

```
auto oxygen::AnimatedMeshComponent::BeginAnimation (
            oxyU32 animHash,
            oxyBool loop = true) -> void
```

6.1.1.2 GetCurrenntAnimationHash()

```
auto oxygen::AnimatedMeshComponent::GetCurrenntAnimationHash () const -> oxyU32   [inline]
```

6.1.1.3 LoadByName()

```
auto oxygen::AnimatedMeshComponent::LoadByName (
            std::string_view name) -> oxyBool
```

6.1.1.4 OXYGENOBJECT()

```
oxygen::AnimatedMeshComponent::OXYGENOBJECT (
            AnimatedMeshComponent ,
            Component )
```

6.1.1.5 Render()

```
auto oxygen::AnimatedMeshComponent::Render () const -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

6.1.1.6 SetLocalOffset()

```
auto oxygen::AnimatedMeshComponent::SetLocalOffset (
            const oxyVec3 & offset) -> void   [inline]
```

6.1.1.7 SetLocalRotation()

```
auto oxygen::AnimatedMeshComponent::SetLocalRotation (
            const oxyQuat & rotation) -> void   [inline]
```

6.1.1.8 Update()

```
auto oxygen::AnimatedMeshComponent::Update (
            float deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/AnimatedMeshComponent/AnimatedMeshComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/AnimatedMeshComponent/AnimatedMeshComponent.cc

6.2 oxygen::AnimatedMeshResource Struct Reference

`#include <AnimatedMeshResource.h>`

Public Attributes

- std::shared_ptr< const struct StaticMeshResource > m_rootPose
- std::unordered_map< oxyU32, AnimationInfo > m_animations

6.2.1 Member Data Documentation

6.2.1.1 m_animations

`std::unordered_map<oxyU32, AnimationInfo> oxygen::AnimatedMeshResource::m_animations`

6.2.1.2 m_rootPose

`std::shared_ptr<const struct StaticMeshResource> oxygen::AnimatedMeshResource::m_rootPose`

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/AnimatedMeshResource.h

6.3 oxygen::AnimationInfo Struct Reference

`#include <AnimatedMeshResource.h>`

Public Attributes

- std::vector< std::vector< oxyVec3 > > m_frames

6.3.1 Member Data Documentation

6.3.1.1 m_frames

`std::vector<std::vector<oxyVec3> > oxygen::AnimationInfo::m_frames`

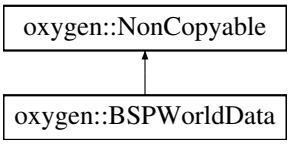
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/AnimatedMeshResource.h

6.4 oxygen::BSPWorldData Struct Reference

`#include <BSP.h>`

Inheritance diagram for oxygen::BSPWorldData:



Public Member Functions

- auto Load (std::string_view mapname) -> oxyBool

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Public Attributes

- std::vector< std::unordered_map< std::string, std::string > > m_entitiesText
- std::vector< BSPDefines::Plane > m_planes
- std::vector< BSPDefines::MipTex > m_miptex
- std::vector< BSPDefines::Vertex > m_vertices
- std::vector< oxyU8 > m_visibility
- std::vector< BSPDefines::Node > m_nodes
- std::vector< BSPDefines::TexInfo > m_texinfo
- std::vector< BSPDefines::Face > m_faces
- std::vector< BSPDefines::ClipNode > m_clipNodes
- std::vector< BSPDefines::Leaf > m_leaves
- std::vector< oxyU16 > m_marksurfaces
- std::vector< BSPDefines::Edge > m_edges
- std::vector< oxyS32 > m_surfedges
- std::vector< BSPDefines::Model > m_models

6.4.1 Member Function Documentation

6.4.1.1 Load()

```
auto oxygen::BSPWorldData::Load (
            std::string_view mapname) -> oxyBool
```

6.4.2 Member Data Documentation

6.4.2.1 m_clipNodes

std::vector<BSPDefines::ClipNode> oxygen::BSPWorldData::m_clipNodes

6.4.2.2 m_edges

std::vector<BSPDefines::Edge> oxygen::BSPWorldData::m_edges

6.4.2.3 m_entitiesText

std::vector<std::unordered_map<std::string, std::string> > oxygen::BSPWorldData::m_entities↩
Text

6.4.2.4 m_faces

std::vector<BSPDefines::Face> oxygen::BSPWorldData::m_faces

6.4.2.5 m_leaves

std::vector<BSPDefines::Leaf> oxygen::BSPWorldData::m_leaves

6.4.2.6 m_marksurfaces

std::vector<oxyU16> oxygen::BSPWorldData::m_marksurfaces

6.4.2.7 m_miptex

std::vector<BSPDefines::MipTex> oxygen::BSPWorldData::m_miptex

6.4.2.8 m_models

std::vector<BSPDefines::Model> oxygen::BSPWorldData::m_models

6.4.2.9 m_nodes

std::vector<BSPDefines::Node> oxygen::BSPWorldData::m_nodes

6.4.2.10 m_planes

std::vector<BSPDefines::Plane> oxygen::BSPWorldData::m_planes

6.4.2.11 m_surfedges

```
std::vector<oxyS32> oxygen::BSPWorldData::m_surfedges
```

6.4.2.12 m_texinfo

```
std::vector<BSPDefines::TexInfo> oxygen::BSPWorldData::m_texinfo
```

6.4.2.13 m_vertices

```
std::vector<BSPDefines::Vertex> oxygen::BSPWorldData::m_vertices
```

6.4.2.14 m_visibility

```
std::vector<oxyU8> oxygen::BSPWorldData::m_visibility
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.cc

6.5 oxygen::CallbackList< TRet, TArgs > Struct Template Reference

```
#include <OxygenTypes.h>
```

Public Types

- using fnptr_type = TRet (*)(void*, TArgs...)

Public Member Functions

- auto AddCallback (std::weak_ptr< void > obj, fnptr_type fn) -> void
- template<typename TFun>
  auto IterateCallbacks (TFun &&fun, TArgs... args) -> void requires(not std::same_as< void, TRet >)
- template<typename TFun>
  auto IterateCallbacks (TFun &&fun, TArgs... args) -> void

6.5.1 Member Typedef Documentation

6.5.1.1 fnptr_type

```
template<typename TRet, typename...  TArgs>
using oxygen::CallbackList< TRet, TArgs >::fnptr_type = TRet (*)(void*, TArgs...)
```

6.5.2 Member Function Documentation

6.5.2.1 AddCallback()

```
template<typename TRet, typename...  TArgs>
auto oxygen::CallbackList< TRet, TArgs >::AddCallback (
             std::weak_ptr< void > obj,
             fnptr_type fn) -> void   [inline]
```

6.5.2.2 IterateCallbacks() [1/2]

```
template<typename TRet, typename...  TArgs>
template<typename TFun>
auto oxygen::CallbackList< TRet, TArgs >::IterateCallbacks (
             TFun && fun,
             TArgs...  args) -> void   [inline]
```

6.5.2.3 IterateCallbacks() [2/2]

```
template<typename TRet, typename...  TArgs>
template<typename TFun>
auto oxygen::CallbackList< TRet, TArgs >::IterateCallbacks (
             TFun && fun,
             TArgs...  args) -> void requires(not std::same_as<void, TRet>)   [inline]
```

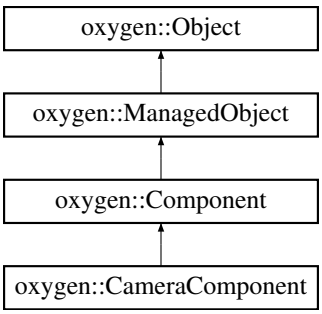
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/OxygenTypes.h

6.6 oxygen::CameraComponent Struct Reference

```
#include <CameraComponent.h>
```

Inheritance diagram for oxygen::CameraComponent:



Public Member Functions

- OXYGENOBJECT (CameraComponent, Component)
- auto GetCameraLocalOffset () const -> const oxyVec3 &
- auto GetEuler () const -> const oxyVec3 &
- auto GetCameraForward () const -> const oxyVec3 &
- auto GetCameraUp () const -> const oxyVec3 &
- auto GetCameraRight () const -> const oxyVec3 &
- auto GetVerticalFov () const -> oxyF32
- auto GetNearClip () const -> oxyF32
- auto GetFarClip () const -> oxyF32
- auto GetViewMatrix () const -> const oxyMat4x4 &
- auto GetProjectionMatrix () const -> const oxyMat4x4 &
- auto GetViewProjectionMatrix () const -> const oxyMat4x4 &
- auto SetEuler (const oxyVec3 &euler) -> void
- auto SetLocalOffset (const oxyVec3 &localOffset) -> void
- auto SetNearClip (oxyF32 nearClip) -> void
- auto SetFarClip (oxyF32 farClip) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (oxyF32 deltaTimeSeconds) -> void override

Protected Member Functions inherited from oxygen::Component

- virtual auto Render () const -> void

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.6.1 Member Function Documentation

6.6.1.1 GetCameraForward()

```
auto oxygen::CameraComponent::GetCameraForward () const -> const oxyVec3&   [inline]
```

6.6.1.2 GetCameraLocalOffset()

```
auto oxygen::CameraComponent::GetCameraLocalOffset () const -> const oxyVec3&   [inline]
```

6.6.1.3 GetCameraRight()

```
auto oxygen::CameraComponent::GetCameraRight () const -> const oxyVec3&   [inline]
```

6.6.1.4 GetCameraUp()

```
auto oxygen::CameraComponent::GetCameraUp () const -> const oxyVec3&   [inline]
```

6.6.1.5 GetEuler()

```
auto oxygen::CameraComponent::GetEuler () const -> const oxyVec3&   [inline]
```

6.6.1.6 GetFarClip()

```
auto oxygen::CameraComponent::GetFarClip () const -> oxyF32   [inline]
```

6.6.1.7 GetNearClip()

```
auto oxygen::CameraComponent::GetNearClip () const -> oxyF32   [inline]
```

6.6.1.8 GetProjectionMatrix()

```
auto oxygen::CameraComponent::GetProjectionMatrix () const -> const oxyMat4x4&   [inline]
```

6.6.1.9 GetVerticalFov()

```
auto oxygen::CameraComponent::GetVerticalFov () const -> oxyF32   [inline]
```

6.6.1.10 GetViewMatrix()

```
auto oxygen::CameraComponent::GetViewMatrix () const -> const oxyMat4x4&   [inline]
```

6.6.1.11 GetViewProjectionMatrix()

```
auto oxygen::CameraComponent::GetViewProjectionMatrix () const -> const oxyMat4x4&   [inline]
```

6.6.1.12 OXYGENOBJECT()

```
oxygen::CameraComponent::OXYGENOBJECT (
            CameraComponent ,
            Component )
```

6.6.1.13 SetEuler()

```
auto oxygen::CameraComponent::SetEuler (
            const oxyVec3 & euler) -> void   [inline]
```

6.6.1.14 SetFarClip()

```
auto oxygen::CameraComponent::SetFarClip (
            oxyF32 farClip) -> void   [inline]
```

6.6.1.15 SetLocalOffset()

```
auto oxygen::CameraComponent::SetLocalOffset (
            const oxyVec3 & localOffset) -> void   [inline]
```

6.6.1.16 SetNearClip()

```
auto oxygen::CameraComponent::SetNearClip (
            oxyF32 nearClip) -> void   [inline]
```

6.6.1.17 Update()

```
auto oxygen::CameraComponent::Update (
            oxyF32 deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/CameraComponent/CameraComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/CameraComponent/CameraComponent.cc

6.7 oxygen::BSPDefines::ClipNode Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyU32 m_planeIndex
- oxyS16 m_children [2]

6.7.1 Member Data Documentation

6.7.1.1 m_children

```
oxyS16 oxygen::BSPDefines::ClipNode::m_children[2]
```

6.7.1.2 m_planeIndex

```
oxyU32 oxygen::BSPDefines::ClipNode::m_planeIndex
```

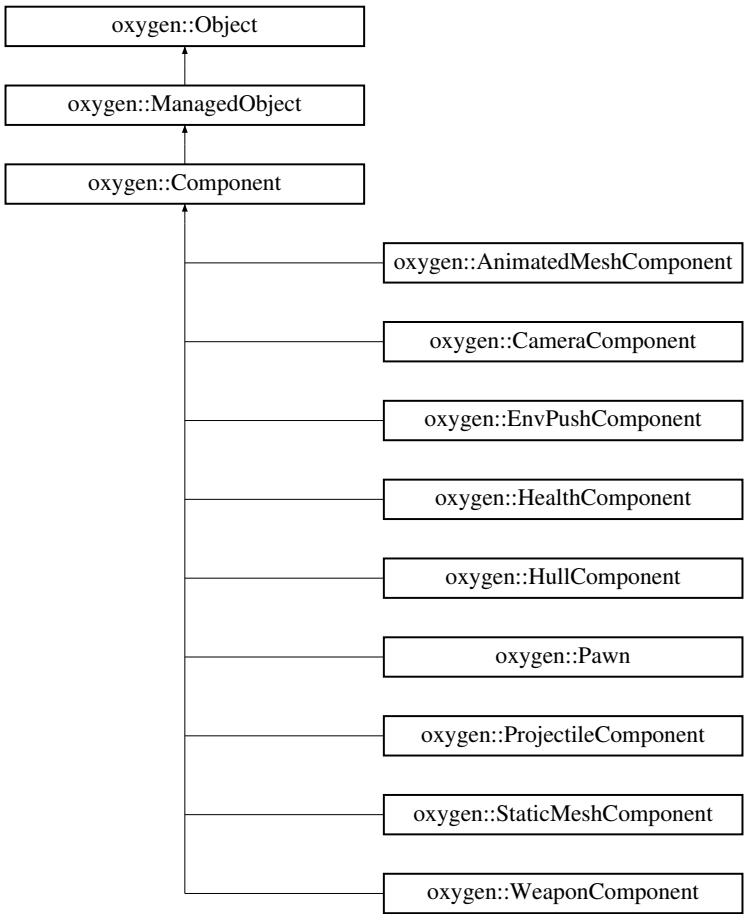
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.8 oxygen::Component Struct Reference

`#include <Component.h>`

Inheritance diagram for oxygen::Component:



Public Member Functions

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- virtual auto Update (oxyF32 deltaTimeSeconds) -> void
- virtual auto Render () const -> void

Friends

- struct Entity

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.8.1 Member Function Documentation

6.8.1.1 GetEntity()

```
auto oxygen::Component::GetEntity () const -> std::shared_ptr<Entity>   [inline]
```

6.8.1.2 IsEnabled()

```
auto oxygen::Component::IsEnabled () const -> oxyBool   [inline]
```

6.8.1.3 OXYGENOBJECT()

```
oxygen::Component::OXYGENOBJECT (
            Component ,
            ManagedObject )
```

6.8.1.4 Render()

```
virtual auto oxygen::Component::Render () const -> void  [inline], [protected], [virtual]
```

Reimplemented in oxygen::AnimatedMeshComponent, oxygen::Pawn, and oxygen::StaticMeshComponent.

6.8.1.5 SetEnabled()

```
auto oxygen::Component::SetEnabled (
            oxyBool enabled) -> void   [inline]
```

6.8.1.6 Update()

```
virtual auto oxygen::Component::Update (
            oxyF32 deltaTimeSeconds) -> void  [inline], [protected], [virtual]
```

Reimplemented in oxygen::AnimatedMeshComponent, oxygen::CameraComponent, oxygen::EnvPushComponent, oxygen::HullComponent, oxygen::Pawn, oxygen::ProjectileComponent, and oxygen::WeaponComponent.

6.8.2 Friends And Related Symbol Documentation

6.8.2.1 Entity

```
friend struct Entity  [friend]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/Component.h

6.9 oxygen::GfxSoftwareRasterizer::CountingIterator< T > Struct Template Reference

Public Types

- using iterator_category = std::random_access_iterator_tag
- using value_type = T
- using difference_type = T
- using pointer = T*
- using reference = T&

Public Member Functions

- const T & operator∗ () const
- CountingIterator & operator++ ()
- CountingIterator operator++ (int)
- CountingIterator & operator-- ()
- CountingIterator operator-- (int)
- CountingIterator & operator+= (const T &lhs)
- CountingIterator & operator-= (const T &lhs)
- CountingIterator operator+ (const T &lhs) const
- CountingIterator operator- (const T &lhs) const
- bool operator== (const CountingIterator &lhs) const
- bool operator!= (const CountingIterator &lhs) const
- const T operator+ (const CountingIterator &lhs) const
- T operator- (const CountingIterator &lhs) const

Public Attributes

- T m_value

6.9.1 Member Typedef Documentation

6.9.1.1 difference_type

```
template<typename T>
using oxygen::GfxSoftwareRasterizer::CountingIterator< T >::difference_type = T
```

6.9.1.2 iterator_category

```
template<typename T>
using oxygen::GfxSoftwareRasterizer::CountingIterator< T >::iterator_category = std::random_↩
access_iterator_tag
```

6.9.1.3 pointer

```
template<typename T>
using oxygen::GfxSoftwareRasterizer::CountingIterator< T >::pointer = T*
```

6.9.1.4 reference

```
template<typename T>
using oxygen::GfxSoftwareRasterizer::CountingIterator< T >::reference = T&
```

6.9.1.5 value_type

```
template<typename T>
using oxygen::GfxSoftwareRasterizer::CountingIterator< T >::value_type = T
```

6.9.2 Member Function Documentation

6.9.2.1 operator"!=()

```
template<typename T>
bool oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator!= (
            const CountingIterator< T > & lhs) const  [inline]
```

6.9.2.2 operator*()

```
template<typename T>
const T & oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator* () const  [inline]
```

6.9.2.3 operator+() [1/2]

```
template<typename T>
const T oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator+ (
            const CountingIterator< T > & lhs) const  [inline]
```

6.9.2.4 operator+() [2/2]

```
template<typename T>
CountingIterator oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator+ (
            const T & lhs) const  [inline]
```

6.9.2.5 operator++() [1/2]

```
template<typename T>
CountingIterator & oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator++ ()  [inline]
```

6.9.2.6 operator++() [2/2]

```
template<typename T>
CountingIterator oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator++ (
            int )  [inline]
```

6.9.2.7 operator+=()

```
template<typename T>
CountingIterator & oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator+= (
            const T & lhs)  [inline]
```

6.9.2.8 operator-() [1/2]

```
template<typename T>
T oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator- (
            const CountingIterator< T > & lhs) const  [inline]
```

6.9.2.9 operator-() [2/2]

```
template<typename T>
CountingIterator oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator- (
            const T & lhs) const  [inline]
```

6.9.2.10 operator--() [1/2]

```
template<typename T>
CountingIterator & oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator-- ()  [inline]
```

6.9.2.11 operator--() [2/2]

```
template<typename T>
CountingIterator oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator-- (
            int )  [inline]
```

6.9.2.12 operator-=()

```
template<typename T>
CountingIterator & oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator-= (
            const T & lhs)  [inline]
```

6.9.2.13 operator==()

```
template<typename T>
bool oxygen::GfxSoftwareRasterizer::CountingIterator< T >::operator== (
            const CountingIterator< T > & lhs) const  [inline]
```

6.9.3 Member Data Documentation

6.9.3.1 m_value

```
template<typename T>
T oxygen::GfxSoftwareRasterizer::CountingIterator< T >::m_value
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxSoftwareRasterize.inl

6.10 oxygen::BSPDefines::Edge Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyU16 m_vertexIndices [2]

6.10.1 Member Data Documentation

6.10.1.1 m_vertexIndices

```
oxyU16 oxygen::BSPDefines::Edge::m_vertexIndices[2]
```

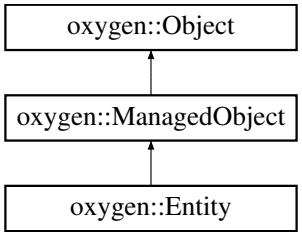
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.11 oxygen::Entity Struct Reference

```
#include <Entity.h>
```

Inheritance diagram for oxygen::Entity:



Public Member Functions

- OXYGENOBJECT (Entity, ManagedObject)
- auto GetLocalPosition () const -> const oxyVec3 &
- auto GetLocalRotation () const -> const oxyQuat &
- auto GetLocalScale () const -> const oxyVec3 &
- auto GetWorldPosition () const -> oxyVec3
- auto GetWorldRotation () const -> oxyQuat
- auto GetWorldScale () const -> oxyVec3
- auto GetWorldTransformMatrix () const -> oxyMat4x4
- auto GetFlag (EntityFlags flag) const -> oxyBool
- auto GetWorld () const -> std::shared_ptr< struct World >
- auto GetParent () const -> std::shared_ptr< Entity >
- auto GetRenderOcclusionMin () const -> const oxyVec3 &
- auto GetRenderOcclusionMax () const -> const oxyVec3 &
- auto SetLocalPosition (const oxyVec3 &position) -> void
- auto SetLocalRotation (const oxyQuat &rotation) -> void
- auto SetLocalScale (const oxyVec3 &scale) -> void
- auto SetWorldPosition (const oxyVec3 &position) -> void
- auto SetWorldRotation (const oxyQuat &rotation) -> void
- auto SetWorldScale (const oxyVec3 &scale) -> void

- auto SetFlag (EntityFlags flag, oxyBool state) -> void
- auto SetRenderOcclusionMin (const oxyVec3 &min) -> void
- auto SetRenderOcclusionMax (const oxyVec3 &max) -> void
- template<typename T>
  auto AddComponent (oxyObjectID id=0) -> std::shared_ptr< T >
- template<typename T>
  auto GetComponent () const -> std::shared_ptr< T >
- auto Destroy () -> void
- auto SetParent (std::shared_ptr< Entity > parent) -> void
- auto Update (oxyF32 deltaTimeSeconds) -> void
- auto Render () const -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Friends

- struct World

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.11.1 Member Function Documentation

6.11.1.1 AddComponent()

```
template<typename T>
auto oxygen::Entity::AddComponent (
            oxyObjectID id = 0) -> std::shared_ptr<T>   [inline]
```

6.11.1.2 Destroy()

```
auto oxygen::Entity::Destroy () -> void
```

6.11.1.3 GetComponent()

```
template<typename T>
auto oxygen::Entity::GetComponent () const -> std::shared_ptr<T>   [inline]
```

6.11.1.4 GetFlag()

```
auto oxygen::Entity::GetFlag (
            EntityFlags flag) const -> oxyBool   [inline]
```

6.11.1.5 GetLocalPosition()

```
auto oxygen::Entity::GetLocalPosition () const -> const oxyVec3&   [inline]
```

6.11.1.6 GetLocalRotation()

```
auto oxygen::Entity::GetLocalRotation () const -> const oxyQuat&   [inline]
```

6.11.1.7 GetLocalScale()

```
auto oxygen::Entity::GetLocalScale () const -> const oxyVec3&   [inline]
```

6.11.1.8 GetParent()

```
auto oxygen::Entity::GetParent () const -> std::shared_ptr<Entity>   [inline]
```

6.11.1.9 GetRenderOcclusionMax()

```
auto oxygen::Entity::GetRenderOcclusionMax () const -> const oxyVec3&   [inline]
```

6.11.1.10 GetRenderOcclusionMin()

```
auto oxygen::Entity::GetRenderOcclusionMin () const -> const oxyVec3&   [inline]
```

6.11.1.11 GetWorld()

```
auto oxygen::Entity::GetWorld () const -> std::shared_ptr<struct World>   [inline]
```

6.11.1.12 GetWorldPosition()

```
auto oxygen::Entity::GetWorldPosition () const -> oxyVec3
```

6.11.1.13 GetWorldRotation()

```
auto oxygen::Entity::GetWorldRotation () const -> oxyQuat
```

6.11.1.14 GetWorldScale()

```
auto oxygen::Entity::GetWorldScale () const -> oxyVec3
```

6.11.1.15 GetWorldTransformMatrix()

```
auto oxygen::Entity::GetWorldTransformMatrix () const -> oxyMat4x4
```

6.11.1.16 OXYGENOBJECT()

```
oxygen::Entity::OXYGENOBJECT (
            Entity ,
            ManagedObject )
```

6.11.1.17 Render()

```
auto oxygen::Entity::Render () const -> void
```

6.11.1.18 SetFlag()

```
auto oxygen::Entity::SetFlag (
            EntityFlags flag,
            oxyBool state) -> void   [inline]
```

6.11.1.19 SetLocalPosition()

```
auto oxygen::Entity::SetLocalPosition (
            const oxyVec3 & position) -> void   [inline]
```

6.11.1.20 SetLocalRotation()

```
auto oxygen::Entity::SetLocalRotation (
            const oxyQuat & rotation) -> void   [inline]
```

6.11.1.21 SetLocalScale()

```
auto oxygen::Entity::SetLocalScale (
            const oxyVec3 & scale) -> void   [inline]
```

6.11.1.22 SetParent()

```
auto oxygen::Entity::SetParent (
            std::shared_ptr< Entity > parent) -> void
```

6.11.1.23 SetRenderOcclusionMax()

```
auto oxygen::Entity::SetRenderOcclusionMax (
            const oxyVec3 & max) -> void   [inline]
```

6.11.1.24 SetRenderOcclusionMin()

```
auto oxygen::Entity::SetRenderOcclusionMin (
            const oxyVec3 & min) -> void   [inline]
```

6.11.1.25 SetWorldPosition()

```
auto oxygen::Entity::SetWorldPosition (
            const oxyVec3 & position) -> void
```

6.11.1.26 SetWorldRotation()

```
auto oxygen::Entity::SetWorldRotation (
            const oxyQuat & rotation) -> void
```

6.11.1.27 SetWorldScale()

```
auto oxygen::Entity::SetWorldScale (
            const oxyVec3 & scale) -> void
```

6.11.1.28 Update()

```
auto oxygen::Entity::Update (
            oxyF32 deltaTimeSeconds) -> void
```

6.11.2 Friends And Related Symbol Documentation

6.11.2.1 World

```
friend struct World  [friend]
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Entity/Entity.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Entity/Entity.cc

6.12 oxygen::EntityHierarchy Struct Reference

```
#include <Entity.h>
```

Public Attributes

- std::shared_ptr< struct Entity > m_parent {}
- std::vector< std::shared_ptr< struct Entity > > m_children
- std::weak_ptr< struct Entity > m_self {}

6.12.1 Member Data Documentation

6.12.1.1 m_children

```
std::vector<std::shared_ptr<struct Entity> > oxygen::EntityHierarchy::m_children
```

6.12.1.2 m_parent

```
std::shared_ptr<struct Entity> oxygen::EntityHierarchy::m_parent {}
```

6.12.1.3 m_self

```
std::weak_ptr<struct Entity> oxygen::EntityHierarchy::m_self {}
```

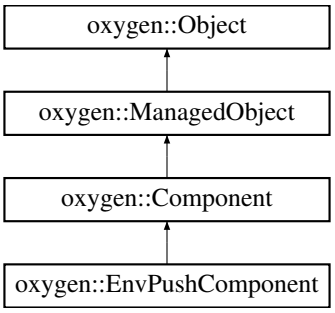
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Entity/Entity.h

6.13 oxygen::EnvPushComponent Struct Reference

`#include <EnvPushComponent.h>`

Inheritance diagram for oxygen::EnvPushComponent:



Public Member Functions

- OXYGENOBJECT (EnvPushComponent, Component)
- auto GetVelocity () const -> const oxyVec3 &
- auto GetRadius () const -> oxyF32
- auto SetVelocity (const oxyVec3 &velocity) -> void
- auto SetRadius (oxyF32 radius) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (oxyF32 deltaTimeSeconds) -> void override

Protected Member Functions inherited from oxygen::Component

- virtual auto Render () const -> void

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.13.1 Member Function Documentation

6.13.1.1 GetRadius()

```
auto oxygen::EnvPushComponent::GetRadius () const -> oxyF32   [inline]
```

6.13.1.2 GetVelocity()

```
auto oxygen::EnvPushComponent::GetVelocity () const -> const oxyVec3&   [inline]
```

6.13.1.3 OXYGENOBJECT()

```
oxygen::EnvPushComponent::OXYGENOBJECT (
            EnvPushComponent ,
            Component )
```

6.13.1.4 SetRadius()

```
auto oxygen::EnvPushComponent::SetRadius (
            oxyF32 radius) -> void   [inline]
```

6.13.1.5 SetVelocity()

```
auto oxygen::EnvPushComponent::SetVelocity (
            const oxyVec3 & velocity) -> void   [inline]
```

6.13.1.6 Update()

```
auto oxygen::EnvPushComponent::Update (
            oxyF32 deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/EnvPushComponent/EnvPushComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/EnvPushComponent/EnvPushComponent.cc

6.14 oxygen::BSPDefines::Face Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyU16 m_planeIndex
- oxyU16 m_side
- oxyU32 m_firstEdgeIndex
- oxyU16 m_edgeCount
- oxyU16 m_texInfoIndex
- oxyU8 m_lightStyles [k_MaxLightMaps]
- oxyU32 m_lightMapOffset

6.14.1 Member Data Documentation

6.14.1.1 m_edgeCount

```
oxyU16 oxygen::BSPDefines::Face::m_edgeCount
```

6.14.1.2 m_firstEdgeIndex

```
oxyU32 oxygen::BSPDefines::Face::m_firstEdgeIndex
```

6.14.1.3 m_lightMapOffset

```
oxyU32 oxygen::BSPDefines::Face::m_lightMapOffset
```

6.14.1.4 m_lightStyles

```
oxyU8 oxygen::BSPDefines::Face::m_lightStyles[k_MaxLightMaps]
```

6.14.1.5 m_planeIndex

```
oxyU16 oxygen::BSPDefines::Face::m_planeIndex
```

6.14.1.6 m_side

```
oxyU16 oxygen::BSPDefines::Face::m_side
```

6.14.1.7 m_texInfoIndex

```
oxyU16 oxygen::BSPDefines::Face::m_texInfoIndex
```

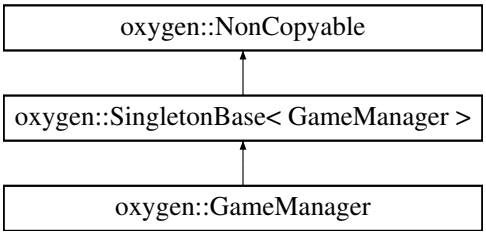
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.15 oxygen::GameManager Struct Reference

```
#include <GameManager.h>
```

Inheritance diagram for oxygen::GameManager:



Public Member Functions

- GameManager ()
- auto Render () -> void
- auto Update (float deltaTimeSeconds) -> void
- auto HostSummonEntity (EntitySpawnType type, const oxyVec3 &pos, const oxyQuat &rot) -> std::shared_ptr< struct Entity >
- auto HostGame (std::string worldName) -> void

Public Member Functions inherited from oxygen::SingletonBase< GameManager >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Friends

- struct NetSystem

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< GameManager >

- static auto GetInstance () -> GameManager &

6.15.1 Constructor & Destructor Documentation

6.15.1.1 GameManager()

```
oxygen::GameManager::GameManager ()
```

6.15.2 Member Function Documentation

6.15.2.1 HostGame()

```
auto oxygen::GameManager::HostGame (
            std::string worldName) -> void
```

6.15.2.2 HostSummonEntity()

```
auto oxygen::GameManager::HostSummonEntity (
            EntitySpawnType type,
            const oxyVec3 & pos,
            const oxyQuat & rot) -> std::shared_ptr<struct Entity>
```

6.15.2.3 Render()

```
auto oxygen::GameManager::Render () -> void
```

6.15.2.4 Update()

```
auto oxygen::GameManager::Update (
            float deltaTimeSeconds) -> void
```

6.15.3 Friends And Related Symbol Documentation

6.15.3.1 NetSystem

```
friend struct NetSystem  [friend]
```

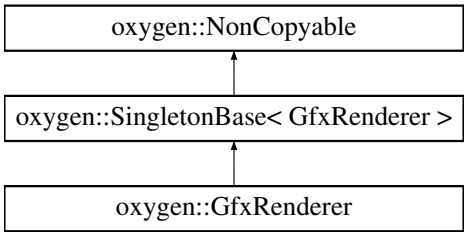
The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/GameManager/GameManager.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/GameManager/GameManager.cc

6.16 oxygen::GfxRenderer Struct Reference

```
#include <GfxRenderer.h>
```

Inheritance diagram for oxygen::GfxRenderer:



Public Member Functions

- GfxRenderer ()
- auto SetViewProjectionMatrix (const oxyMat4x4 &viewProjectionMatrix) -> void
- auto GetViewProjectionMatrix () const -> const oxyMat4x4 &
- auto GetWidth () const -> oxyS32
- auto GetHeight () const -> oxyS32
- auto LoadTexture (std::string_view texturePath) -> std::shared_ptr< const GfxTexture >
- auto OverlayText (std::string_view text, oxyF32 blxndc, oxyF32 blyndc, const oxyVec3 &colour, oxyF32 spacing, oxyF32 size, oxyBool center) -> void
- auto OverlayRect (const oxyVec3 &col, const oxyVec2 &minndc, const oxyVec2 &maxndc) -> void
- auto BeginFrame (oxyS32 w, oxyS32 h) -> void
- auto EndFrame () -> void
- auto SubmitTriToQueue (const GfxTri &tri, GfxRenderStrategy mode, oxyF32 zmult=1.0f) -> void

Public Member Functions inherited from oxygen::SingletonBase< GfxRenderer >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< GfxRenderer >

- static auto GetInstance () -> GfxRenderer &

6.16.1 Constructor & Destructor Documentation

6.16.1.1 GfxRenderer()

```
oxygen::GfxRenderer::GfxRenderer ()
```

6.16.2 Member Function Documentation

6.16.2.1 BeginFrame()

```
auto oxygen::GfxRenderer::BeginFrame (
            oxyS32 w,
            oxyS32 h) -> void
```

6.16.2.2 EndFrame()

```
auto oxygen::GfxRenderer::EndFrame () -> void
```

6.16.2.3 GetHeight()

```
auto oxygen::GfxRenderer::GetHeight () const -> oxyS32   [inline]
```

6.16.2.4 GetViewProjectionMatrix()

```
auto oxygen::GfxRenderer::GetViewProjectionMatrix () const -> const oxyMat4x4&   [inline]
```

6.16.2.5 GetWidth()

```
auto oxygen::GfxRenderer::GetWidth () const -> oxyS32   [inline]
```

6.16.2.6 LoadTexture()

```
auto oxygen::GfxRenderer::LoadTexture (
            std::string_view texturePath) -> std::shared_ptr<const GfxTexture>
```

6.16.2.7 OverlayRect()

```
auto oxygen::GfxRenderer::OverlayRect (
            const oxyVec3 & col,
            const oxyVec2 & minndc,
            const oxyVec2 & maxndc) -> void
```

6.16.2.8 OverlayText()

```
auto oxygen::GfxRenderer::OverlayText (
            std::string_view text,
            oxyF32 blxndc,
            oxyF32 blyndc,
            const oxyVec3 & colour,
            oxyF32 spacing,
            oxyF32 size,
            oxyBool center) -> void
```

6.16.2.9 SetViewProjectionMatrix()

```
auto oxygen::GfxRenderer::SetViewProjectionMatrix (
            const oxyMat4x4 & viewProjectionMatrix) -> void   [inline]
```

6.16.2.10 SubmitTriToQueue()

```
auto oxygen::GfxRenderer::SubmitTriToQueue (
            const GfxTri & tri,
            GfxRenderStrategy mode,
            oxyF32 zmult = 1.0f) -> void
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.cc

6.17 oxygen::GfxTexture Struct Reference

```
#include <GfxRenderer.h>
```

Public Attributes

- oxyU32 m_width
- oxyU32 m_height
- std::string m_texturePath
- std::shared_ptr< const GraphicsAbstraction::Texture > m_texture

6.17.1 Member Data Documentation

6.17.1.1 m_height

```
oxyU32 oxygen::GfxTexture::m_height
```

6.17.1.2 m_texture

```
std::shared_ptr<const GraphicsAbstraction::Texture> oxygen::GfxTexture::m_texture
```

6.17.1.3 m_texturePath

```
std::string oxygen::GfxTexture::m_texturePath
```

6.17.1.4 m_width

```
oxyU32 oxygen::GfxTexture::m_width
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.h

6.18 oxygen::GfxTri Struct Reference

```
#include <GfxRenderer.h>
```

Public Attributes

- GfxVertex m_vertices [3]
- oxyVec3 m_colour
- const GfxTexture * m_texture {}
- GfxCullType m_cullType

6.18.1 Member Data Documentation

6.18.1.1 m_colour

```
oxyVec3 oxygen::GfxTri::m_colour
```

6.18.1.2 m_cullType

```
GfxCullType oxygen::GfxTri::m_cullType
```

6.18.1.3 m_texture

```
const GfxTexture* oxygen::GfxTri::m_texture {}
```

6.18.1.4 m_vertices

```
GfxVertex oxygen::GfxTri::m_vertices[3]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.h

6.19 oxygen::GfxVertex Struct Reference

```
#include <GfxRenderer.h>
```

Public Attributes

- oxyVec4 m_position
- oxyVec2 m_uv

6.19.1 Member Data Documentation

6.19.1.1 m_position

```
oxyVec4 oxygen::GfxVertex::m_position
```

6.19.1.2 m_uv

```
oxyVec2 oxygen::GfxVertex::m_uv
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.h

6.20 oxygen::BSPDefines::Header Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyS32 m_version
- Lump m_lumps [LumpIndex_Count]

6.20.1 Member Data Documentation

6.20.1.1 m_lumps

```
Lump oxygen::BSPDefines::Header::m_lumps[LumpIndex_Count]
```

6.20.1.2 m_version

```
oxyS32 oxygen::BSPDefines::Header::m_version
```

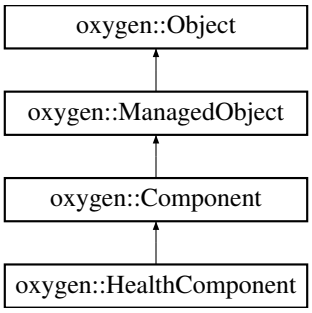
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.21 oxygen::HealthComponent Struct Reference

```
#include <HealthComponent.h>
```

Inheritance diagram for oxygen::HealthComponent:



Public Member Functions

- OXYGENOBJECT (HealthComponent, Component)
- template<typename... TArgs>
  auto AddHealthStateChangedEvent (TArgs &&... args) -> void
- template<typename... TArgs>
  auto AddHealedEvent (TArgs &&... args) -> void
- template<typename... TArgs>
  auto AddDamagedEvent (TArgs &&... args) -> void
- auto Heal (oxyS32 amount) -> void
- auto Damage (oxyS32 amount, DamageType type) -> void
- auto SetHealth (oxyU32 health) -> void
- auto SetMaxHealth (oxyU32 maxHealth) -> void
- auto GetHealth () const -> oxyU32
- auto GetMaxHealth () const -> oxyU32
- auto GetState () const -> HealthState

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Friends

- struct GameManager

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

Protected Member Functions inherited from oxygen::Component

- virtual auto Update (oxyF32 deltaTimeSeconds) -> void
- virtual auto Render () const -> void

6.21.1 Member Function Documentation

6.21.1.1 AddDamagedEvent()

```
template<typename... TArgs>
auto oxygen::HealthComponent::AddDamagedEvent (
            TArgs &&... args) -> void   [inline]
```

6.21.1.2 AddHealedEvent()

```
template<typename... TArgs>
auto oxygen::HealthComponent::AddHealedEvent (
            TArgs &&... args) -> void   [inline]
```

6.21.1.3 AddHealthStateChangedEvent()

```
template<typename... TArgs>
auto oxygen::HealthComponent::AddHealthStateChangedEvent (
            TArgs &&... args) -> void   [inline]
```

6.21.1.4 Damage()

```
auto oxygen::HealthComponent::Damage (
            oxyS32 amount,
            DamageType type) -> void
```

6.21.1.5 GetHealth()

```
auto oxygen::HealthComponent::GetHealth () const -> oxyU32   [inline]
```

6.21.1.6 GetMaxHealth()

```
auto oxygen::HealthComponent::GetMaxHealth () const -> oxyU32   [inline]
```

6.21.1.7 GetState()

```
auto oxygen::HealthComponent::GetState () const -> HealthState   [inline]
```

6.21.1.8 Heal()

```
auto oxygen::HealthComponent::Heal (
            oxyS32 amount) -> void
```

6.21.1.9 OXYGENOBJECT()

```
oxygen::HealthComponent::OXYGENOBJECT (
            HealthComponent ,
            Component )
```

6.21.1.10 SetHealth()

```
auto oxygen::HealthComponent::SetHealth (
            oxyU32 health) -> void   [inline]
```

6.21.1.11 SetMaxHealth()

```
auto oxygen::HealthComponent::SetMaxHealth (
            oxyU32 maxHealth) -> void   [inline]
```

6.21.2 Friends And Related Symbol Documentation

6.21.2.1 GameManager

```
friend struct GameManager  [friend]
```

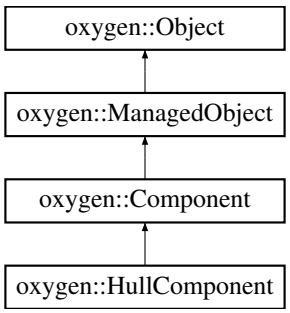
The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/HealthComponent/HealthComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/HealthComponent/HealthComponent.cc

6.22 oxygen::HullComponent Struct Reference

```
#include <HullComponent.h>
```

Inheritance diagram for oxygen::HullComponent:



Public Member Functions

- OXYGENOBJECT (HullComponent, Component)
- template<typename... TArgs>
  auto AddCollideEvent (TArgs &&... args) -> void
- template<typename... TArgs>
  auto AddBounceEvent (TArgs &&... args) -> void
- auto TraceLine (const oxyVec3 &start, const oxyVec3 &end, oxyVec3 &outPosition, oxyVec3 &outNormal) const -> oxyBool
- auto CollidesWithHull (const oxyVec3 &otherHullWorldPosition, CollisionHull otherHull, oxyVec3 &outPosition, oxyVec3 &outNormal) const -> oxyBool
- auto IsWithinRadius (const oxyVec3 &position, oxyF32 radius) const -> oxyBool
- auto DoesIgnoreEntity (const struct Entity *entity) const -> oxyBool
- auto GetHull () const -> CollisionHull
- auto GetVelocity () const -> const oxyVec3 &
- auto GetGravityPerSecond () const -> oxyF32
- auto GetDrag () const -> oxyF32
- auto GetSolidToOtherHulls () const -> oxyBool
- auto GetBounceVelocityMultiplier () const -> oxyF32
- auto GetResponseType () const -> oxyBool
- auto SetHull (CollisionHull hull) -> void
- auto SetVelocity (const oxyVec3 &velocity) -> void
- auto SetGravityPerSecond (oxyF32 gravityPerSecond) -> void
- auto SetDrag (oxyF32 drag) -> void
- auto SetSolidToOtherHulls (oxyBool solid) -> void
- auto SetBounceVelocityMultiplier (oxyF32 bounceVelocityMultiplier) -> void
- auto SetResponse (CollisionResponseType response) -> void
- auto AddToIgnoreList (const std::shared_ptr< struct Entity > &entity) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (oxyF32 deltaTimeSeconds) -> void override

Protected Member Functions inherited from oxygen::Component

- virtual auto Render () const -> void

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.22.1 Member Function Documentation

6.22.1.1 AddBounceEvent()

```
template<typename... TArgs>
auto oxygen::HullComponent::AddBounceEvent (
            TArgs &&... args) -> void   [inline]
```

6.22.1.2 AddCollideEvent()

```
template<typename... TArgs>
auto oxygen::HullComponent::AddCollideEvent (
            TArgs &&... args) -> void   [inline]
```

6.22.1.3 AddToIgnoreList()

```
auto oxygen::HullComponent::AddToIgnoreList (
            const std::shared_ptr< struct Entity > & entity) -> void
```

6.22.1.4 CollidesWithHull()

```
auto oxygen::HullComponent::CollidesWithHull (
            const oxyVec3 & otherHullWorldPosition,
            CollisionHull otherHull,
            oxyVec3 & outPosition,
            oxyVec3 & outNormal) const -> oxyBool
```

6.22.1.5 DoesIgnoreEntity()

```
auto oxygen::HullComponent::DoesIgnoreEntity (
            const struct Entity * entity) const -> oxyBool
```

6.22.1.6 GetBounceVelocityMultiplier()

```
auto oxygen::HullComponent::GetBounceVelocityMultiplier () const -> oxyF32   [inline]
```

6.22.1.7 GetDrag()

```
auto oxygen::HullComponent::GetDrag () const -> oxyF32   [inline]
```

6.22.1.8 GetGravityPerSecond()

```
auto oxygen::HullComponent::GetGravityPerSecond () const -> oxyF32   [inline]
```

6.22.1.9 GetHull()

```
auto oxygen::HullComponent::GetHull () const -> CollisionHull   [inline]
```

6.22.1.10 GetResponseType()

```
auto oxygen::HullComponent::GetResponseType () const -> oxyBool   [inline]
```

6.22.1.11 GetSolidToOtherHulls()

```
auto oxygen::HullComponent::GetSolidToOtherHulls () const -> oxyBool   [inline]
```

6.22.1.12 GetVelocity()

```
auto oxygen::HullComponent::GetVelocity () const -> const oxyVec3&   [inline]
```

6.22.1.13 IsWithinRadius()

```
auto oxygen::HullComponent::IsWithinRadius (
            const oxyVec3 & position,
            oxyF32 radius) const -> oxyBool
```

6.22.1.14 OXYGENOBJECT()

```
oxygen::HullComponent::OXYGENOBJECT (
            HullComponent ,
            Component )
```

6.22.1.15 SetBounceVelocityMultiplier()

```
auto oxygen::HullComponent::SetBounceVelocityMultiplier (
            oxyF32 bounceVelocityMultiplier) -> void   [inline]
```

6.22.1.16 SetDrag()

```
auto oxygen::HullComponent::SetDrag (
            oxyF32 drag) -> void   [inline]
```

6.22.1.17 SetGravityPerSecond()

```
auto oxygen::HullComponent::SetGravityPerSecond (
            oxyF32 gravityPerSecond) -> void   [inline]
```

6.22.1.18 SetHull()

```
auto oxygen::HullComponent::SetHull (
            CollisionHull hull) -> void   [inline]
```

6.22.1.19 SetResponse()

```
auto oxygen::HullComponent::SetResponse (
            CollisionResponseType response) -> void   [inline]
```

6.22.1.20 SetSolidToOtherHulls()

```
auto oxygen::HullComponent::SetSolidToOtherHulls (
            oxyBool solid) -> void   [inline]
```

6.22.1.21 SetVelocity()

```
auto oxygen::HullComponent::SetVelocity (
            const oxyVec3 & velocity) -> void   [inline]
```

6.22.1.22 TraceLine()

```
auto oxygen::HullComponent::TraceLine (
            const oxyVec3 & start,
            const oxyVec3 & end,
            oxyVec3 & outPosition,
            oxyVec3 & outNormal) const -> oxyBool
```

6.22.1.23 Update()

```
auto oxygen::HullComponent::Update (
            oxyF32 deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

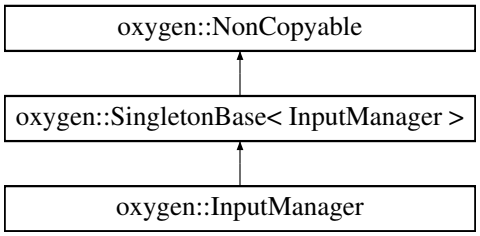
The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/HullComponent/HullComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/HullComponent/HullComponent.cc

6.23 oxygen::InputManager Struct Reference

```
#include <InputManager.h>
```

Inheritance diagram for oxygen::InputManager:



Public Member Functions

- auto GetMousePosition () const -> const oxyVec2
- auto GetMouseDelta () const -> const oxyVec2
- auto IsKeyDown (KeyboardButton key) const -> oxyBool
- auto WasKeyDown (KeyboardButton key) const -> oxyBool
- auto IsMouseButtonDown (MouseButton button) const -> oxyBool
- auto WasMouseButtonDown (MouseButton button) const -> oxyBool
- auto IsControllerConnected (oxyU8 controller) const -> oxyBool
- auto IsControllerButtonDown (oxyU8 controller, ControllerButton button) const -> oxyBool
- auto WasControllerButtonDown (oxyU8 controller, ControllerButton button) const -> oxyBool
- auto GetControllerAxis (oxyU8 controller, ControllerAxis axis) const -> oxyF32
- auto GetPreviousControllerAxis (oxyU8 controller, ControllerAxis axis) const -> oxyF32
- auto SetCursorLock (oxyBool lock) -> void
- auto Update () -> void

Public Member Functions inherited from oxygen::SingletonBase< InputManager >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< InputManager >

- static auto GetInstance () -> InputManager &

6.23.1 Member Function Documentation

6.23.1.1 GetControllerAxis()

```
auto oxygen::InputManager::GetControllerAxis (
            oxyU8 controller,
            ControllerAxis axis) const -> oxyF32   [inline]
```

6.23.1.2 GetMouseDelta()

```
auto oxygen::InputManager::GetMouseDelta () const -> const oxyVec2   [inline]
```

6.23.1.3 GetMousePosition()

```
auto oxygen::InputManager::GetMousePosition () const -> const oxyVec2   [inline]
```

6.23.1.4 GetPreviousControllerAxis()

```
auto oxygen::InputManager::GetPreviousControllerAxis (
            oxyU8 controller,
            ControllerAxis axis) const -> oxyF32   [inline]
```

6.23.1.5 IsControllerButtonDown()

```
auto oxygen::InputManager::IsControllerButtonDown (
            oxyU8 controller,
            ControllerButton button) const -> oxyBool   [inline]
```

6.23.1.6 IsControllerConnected()

```
auto oxygen::InputManager::IsControllerConnected (
            oxyU8 controller) const -> oxyBool   [inline]
```

6.23.1.7 IsKeyDown()

```
auto oxygen::InputManager::IsKeyDown (
            KeyboardButton key) const -> oxyBool   [inline]
```

6.23.1.8 IsMouseButtonDown()

```
auto oxygen::InputManager::IsMouseButtonDown (
            MouseButton button) const -> oxyBool   [inline]
```

6.23.1.9 SetCursorLock()

```
auto oxygen::InputManager::SetCursorLock (
            oxyBool lock) -> void   [inline]
```

6.23.1.10 Update()

```
auto oxygen::InputManager::Update () -> void
```

6.23.1.11 WasControllerButtonDown()

```
auto oxygen::InputManager::WasControllerButtonDown (
            oxyU8 controller,
            ControllerButton button) const -> oxyBool   [inline]
```

6.23.1.12 WasKeyDown()

```
auto oxygen::InputManager::WasKeyDown (
            KeyboardButton key) const -> oxyBool   [inline]
```

6.23.1.13 WasMouseButtonDown()

```
auto oxygen::InputManager::WasMouseButtonDown (
            MouseButton button) const -> oxyBool   [inline]
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Input/InputManager.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Input/InputManager.cc

6.24 oxygen::InternalEngineSingletonsOrder Struct Reference

`#include <EngineSingletons.h>`

Public Attributes

- SingletonInstance< ObjectManager > m_objectManagerInstance {}
- SingletonInstance< InputManager > m_inputManagerInstance {}
- SingletonInstance< GfxRenderer > m_gfxRendererInstance {}
- SingletonInstance< ResourceManager > m_resourceManagerInstance {}
- SingletonInstance< UIManager > m_uiManagerInstance {}
- SingletonInstance< NetSystem > m_netSystemInstance {}
- SingletonInstance< GameManager > m_gameManagerInstance {}

6.24.1 Member Data Documentation

6.24.1.1 m_gameManagerInstance

```
SingletonInstance<GameManager> oxygen::InternalEngineSingletonsOrder::m_gameManagerInstance {}
```

6.24.1.2 m_gfxRendererInstance

```
SingletonInstance<GfxRenderer> oxygen::InternalEngineSingletonsOrder::m_gfxRendererInstance {}
```

6.24.1.3 m_inputManagerInstance

```
SingletonInstance<InputManager> oxygen::InternalEngineSingletonsOrder::m_inputManagerInstance
{}
```

6.24.1.4 m_netSystemInstance

```
SingletonInstance<NetSystem> oxygen::InternalEngineSingletonsOrder::m_netSystemInstance {}
```

6.24.1.5 m_objectManagerInstance

```
SingletonInstance<ObjectManager> oxygen::InternalEngineSingletonsOrder::m_objectManager↩
Instance {}
```

6.24.1.6 m_resourceManagerInstance

```
SingletonInstance<ResourceManager> oxygen::InternalEngineSingletonsOrder::m_resourceManager↩
Instance {}
```

6.24.1.7 m_uiManagerInstance

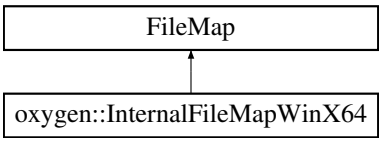
```
SingletonInstance<UIManager> oxygen::InternalEngineSingletonsOrder::m_uiManagerInstance {}
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Singleton/EngineSingletons.h

6.25 oxygen::InternalFileMapWinX64 Struct Reference

Inheritance diagram for oxygen::InternalFileMapWinX64:



Friends

- struct InternalFileMapDeleter
- auto CreateFileMap (std::string_view path, oxyBool write, oxySize requestSize) -> UniqueFileMap

6.25.1 Friends And Related Symbol Documentation

6.25.1.1 CreateFileMap

```
auto CreateFileMap (
            std::string_view path,
            oxyBool write,
            oxySize requestSize) -> UniqueFileMap  [friend]
```

6.25.1.2 InternalFileMapDeleter

```
friend struct InternalFileMapDeleter  [friend]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Platform/PlatformWin64/Platform.cc

6.26 oxygen::BSPDefines::Leaf Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyS32 m_contents
- oxyS32 m_visOffset
- oxyS16 m_mins [3]
- oxyS16 m_maxs [3]
- oxyU16 m_firstMarkSurfaceIndex
- oxyU16 m_markSurfaceCount
- oxyU8 m_ambientLevels [k_NumAmbients]

6.26.1 Member Data Documentation

6.26.1.1 m_ambientLevels

```
oxyU8 oxygen::BSPDefines::Leaf::m_ambientLevels[k_NumAmbients]
```

6.26.1.2 m_contents

```
oxyS32 oxygen::BSPDefines::Leaf::m_contents
```

6.26.1.3 m_firstMarkSurfaceIndex

```
oxyU16 oxygen::BSPDefines::Leaf::m_firstMarkSurfaceIndex
```

6.26.1.4 m_markSurfaceCount

```
oxyU16 oxygen::BSPDefines::Leaf::m_markSurfaceCount
```

6.26.1.5 m_maxs

```
oxyS16 oxygen::BSPDefines::Leaf::m_maxs[3]
```

6.26.1.6 m_mins

```
oxyS16 oxygen::BSPDefines::Leaf::m_mins[3]
```

6.26.1.7 m_visOffset

```
oxyS32 oxygen::BSPDefines::Leaf::m_visOffset
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.27 oxygen::World::LineTraceResult Struct Reference

`#include <World.h>`

Public Attributes

- oxyBool m_allSolid {}
- oxyBool m_startSolid {}
- oxyF32 m_fraction {}
- oxyVec3 m_endPos {}
- oxyVec3 m_planeNormal {}
- oxyF32 m_planeDist {}
- std::shared_ptr< struct Entity > m_hitEntity {}

6.27.1 Member Data Documentation

6.27.1.1 m_allSolid

```
oxyBool oxygen::World::LineTraceResult::m_allSolid {}
```

6.27.1.2 m_endPos

```
oxyVec3 oxygen::World::LineTraceResult::m_endPos {}
```

6.27.1.3 m_fraction

```
oxyF32 oxygen::World::LineTraceResult::m_fraction {}
```

6.27.1.4 m_hitEntity

```
std::shared_ptr<struct Entity> oxygen::World::LineTraceResult::m_hitEntity {}
```

6.27.1.5 m_planeDist

```
oxyF32 oxygen::World::LineTraceResult::m_planeDist {}
```

6.27.1.6 m_planeNormal

```
oxyVec3 oxygen::World::LineTraceResult::m_planeNormal {}
```

6.27.1.7 m_startSolid

```
oxyBool oxygen::World::LineTraceResult::m_startSolid {}
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/World.h

6.28 oxygen::BSPDefines::Lump Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyU32 m_fileOffset
- oxyU32 m_length

6.28.1 Member Data Documentation

6.28.1.1 m_fileOffset

```
oxyU32 oxygen::BSPDefines::Lump::m_fileOffset
```

6.28.1.2 m_length

```
oxyU32 oxygen::BSPDefines::Lump::m_length
```

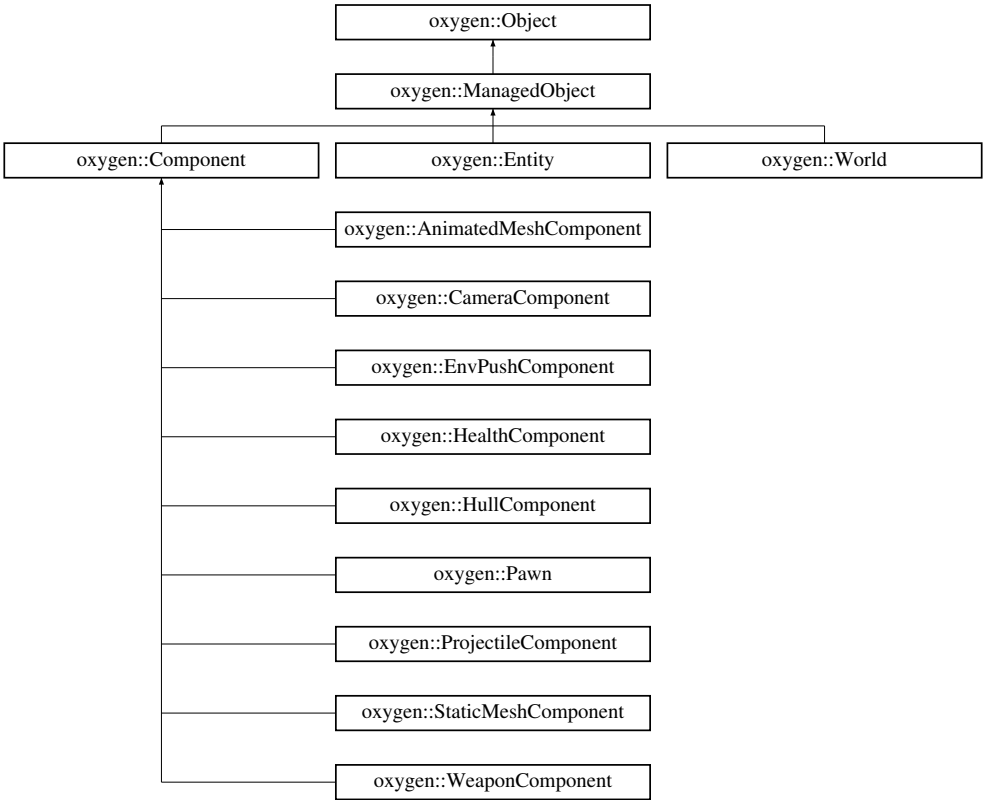
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.29 oxygen::ManagedObject Struct Reference

`#include <ManagedObject.h>`

Inheritance diagram for oxygen::ManagedObject:



Public Member Functions

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Friends

- struct ObjectManager

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.29.1 Member Function Documentation

6.29.1.1 GetHardRef()

```
template<typename RefType>
requires std::is_base_of_v<ManagedObject, RefType>
auto oxygen::ManagedObject::GetHardRef () const -> std::shared_ptr<RefType>   [inline]
```

6.29.1.2 GetObjectID()

```
auto oxygen::ManagedObject::GetObjectID () const -> oxyObjectID   [inline]
```

6.29.1.3 OXYGENOBJECT()

```
oxygen::ManagedObject::OXYGENOBJECT (
            ManagedObject ,
            Object )
```

6.29.2 Friends And Related Symbol Documentation

6.29.2.1 ObjectManager

```
friend struct ObjectManager  [friend]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/ManagedObject.h

6.30 oxygen::BSPDefines::MipTex Struct Reference

`#include <BSP.h>`

Public Attributes

- oxyChar m_name [16]
- oxyU32 m_width
- oxyU32 m_height
- oxyU32 m_offsets [k_NumMipLevels]

6.30.1 Member Data Documentation

6.30.1.1 m_height

```
oxyU32 oxygen::BSPDefines::MipTex::m_height
```

6.30.1.2 m_name

```
oxyChar oxygen::BSPDefines::MipTex::m_name[16]
```

6.30.1.3 m_offsets

```
oxyU32 oxygen::BSPDefines::MipTex::m_offsets[k_NumMipLevels]
```

6.30.1.4 m_width

```
oxyU32 oxygen::BSPDefines::MipTex::m_width
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.31 oxygen::BSPDefines::MipTexLump Struct Reference

`#include <BSP.h>`

Public Attributes

- oxyU32 m_numMipTex
- oxyU32 m_dataOffsets [4]

6.31.1 Member Data Documentation

6.31.1.1 m_dataOffsets

```
oxyU32 oxygen::BSPDefines::MipTexLump::m_dataOffsets[4]
```

6.31.1.2 m_numMipTex

```
oxyU32 oxygen::BSPDefines::MipTexLump::m_numMipTex
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.32 oxygen::BSPDefines::Model Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyF32 m_mins [3]
- oxyF32 m_maxs [3]
- oxyF32 m_origin [3]
- oxyU32 m_headNodes [k_MaxMapHulls]
- oxyU32 m_visLeafs
- oxyU32 m_firstFaceIndex
- oxyU32 m_faceCount

6.32.1 Member Data Documentation

6.32.1.1 m_faceCount

```
oxyU32 oxygen::BSPDefines::Model::m_faceCount
```

6.32.1.2 m_firstFaceIndex

```
oxyU32 oxygen::BSPDefines::Model::m_firstFaceIndex
```

6.32.1.3 m_headNodes

```
oxyU32 oxygen::BSPDefines::Model::m_headNodes[k_MaxMapHulls]
```

6.32.1.4 m_maxs

```
oxyF32 oxygen::BSPDefines::Model::m_maxs[3]
```

6.32.1.5 m_mins

```
oxyF32 oxygen::BSPDefines::Model::m_mins[3]
```

6.32.1.6 m_origin

```
oxyF32 oxygen::BSPDefines::Model::m_origin[3]
```

6.32.1.7 m_visLeafs

```
oxyU32 oxygen::BSPDefines::Model::m_visLeafs
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.33 oxygen::NetConnection Struct Reference

```
#include <NetSystem.h>
```

Public Member Functions

- auto WriteData (oxyU16 type, std::span< const oxyU8 > data) -> void
- auto GetUniqueID () const -> oxyU64

Friends

- struct NetSystem

6.33.1 Member Function Documentation

6.33.1.1 GetUniqueID()

```
auto oxygen::NetConnection::GetUniqueID () const -> oxyU64   [inline]
```

6.33.1.2 WriteData()

```
auto oxygen::NetConnection::WriteData (
            oxyU16 type,
            std::span< const oxyU8 > data) -> void
```

6.33.2 Friends And Related Symbol Documentation

6.33.2.1 NetSystem

```
friend struct NetSystem  [friend]
```

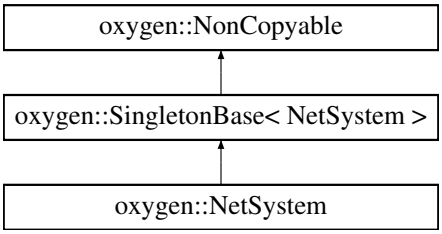
The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Net/NetSystem.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Net/NetSystem.cc

6.34 oxygen::NetSystem Struct Reference

```
#include <NetSystem.h>
```

Inheritance diagram for oxygen::NetSystem:



Public Member Functions

- NetSystem ()
- ~NetSystem ()
- auto Update (oxyF32 deltaTimeSeconds) -> void
- auto StartHost () -> void
- auto ConnectToHost (const std::string &ip) -> oxyBool
- auto KillConnections () -> void
- auto HostSendToAll (oxyU16 type, const std::span< oxyU8 > &data) -> void
- auto HostSendToAllExcept (oxyU64 excludeClientID, oxyU16 type, const std::span< oxyU8 > &data) -> void
- auto CliSendToHost (oxyU16 type, const std::span< oxyU8 > &data) -> void
- auto CliDiscoverHosts () -> void
- auto CliGetDiscoveredHosts () const -> std::span< const std::string >
- auto CliIsDiscoveringHosts () const -> oxyBool
- auto GetNewNetObjID () -> oxyObjectID
- auto IsHost () const -> oxyBool
- auto IsClient () const -> oxyBool

Public Member Functions inherited from oxygen::SingletonBase< NetSystem >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Static Public Attributes

- static constexpr auto k_enginePort = 28672
- static constexpr auto k_engineBroadcastPort = 28678
- static constexpr auto k_timeBetweenPing = 4.0f
- static constexpr auto k_netSessionDefaultMinObjid

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< NetSystem >

- static auto GetInstance () -> NetSystem &

6.34.1 Constructor & Destructor Documentation

6.34.1.1 NetSystem()

```
oxygen::NetSystem::NetSystem ()
```

6.34.1.2 ~NetSystem()

```
oxygen::NetSystem::~NetSystem ()
```

6.34.2 Member Function Documentation

6.34.2.1 CliDiscoverHosts()

```
auto oxygen::NetSystem::CliDiscoverHosts () -> void
```

6.34.2.2 CliGetDiscoveredHosts()

```
auto oxygen::NetSystem::CliGetDiscoveredHosts () const -> std::span<const std::string>   [inline]
```

6.34.2.3 CliIsDiscoveringHosts()

```
auto oxygen::NetSystem::CliIsDiscoveringHosts () const -> oxyBool   [inline]
```

6.34.2.4 CliSendToHost()

```
auto oxygen::NetSystem::CliSendToHost (
            oxyU16 type,
            const std::span< oxyU8 > & data) -> void
```

6.34.2.5 ConnectToHost()

```
auto oxygen::NetSystem::ConnectToHost (
            const std::string & ip) -> oxyBool
```

6.34.2.6 GetNewNetObjID()

```
auto oxygen::NetSystem::GetNewNetObjID () -> oxyObjectID   [inline]
```

6.34.2.7 HostSendToAll()

```
auto oxygen::NetSystem::HostSendToAll (
            oxyU16 type,
            const std::span< oxyU8 > & data) -> void
```

6.34.2.8 HostSendToAllExcept()

```
auto oxygen::NetSystem::HostSendToAllExcept (
            oxyU64 excludeClientID,
            oxyU16 type,
            const std::span< oxyU8 > & data) -> void
```

6.34.2.9 IsClient()

```
auto oxygen::NetSystem::IsClient () const -> oxyBool   [inline]
```

6.34.2.10 IsHost()

```
auto oxygen::NetSystem::IsHost () const -> oxyBool   [inline]
```

6.34.2.11 KillConnections()

```
auto oxygen::NetSystem::KillConnections () -> void
```

6.34.2.12 StartHost()

```
auto oxygen::NetSystem::StartHost () -> void
```

6.34.2.13 Update()

```
auto oxygen::NetSystem::Update (
            oxyF32 deltaTimeSeconds) -> void
```

6.34.3 Member Data Documentation

6.34.3.1 k_engineBroadcastPort

```
auto oxygen::NetSystem::k_engineBroadcastPort = 28678  [inline], [static], [constexpr]
```

6.34.3.2 k_enginePort

```
auto oxygen::NetSystem::k_enginePort = 28672  [inline], [static], [constexpr]
```

6.34.3.3 k_netSessionDefaultMinObjid

```
auto oxygen::NetSystem::k_netSessionDefaultMinObjid  [inline], [static], [constexpr]
```

Initial value:

```
=
           oxyObjectID{0xC0000001}
```

6.34.3.4 k_timeBetweenPing

```
auto oxygen::NetSystem::k_timeBetweenPing = 4.0f  [inline], [static], [constexpr]
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Net/NetSystem.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Net/NetSystem.cc

6.35 oxygen::BSPDefines::Node Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyU32 m_planeIndex
- oxyS16 m_children [2]
- oxyS16 m_mins [3]
- oxyS16 m_maxs [3]
- oxyU16 m_firstFaceIndex
- oxyU16 m_faceCount

6.35.1 Member Data Documentation

6.35.1.1 m_children

```
oxyS16 oxygen::BSPDefines::Node::m_children[2]
```

6.35.1.2 m_faceCount

```
oxyU16 oxygen::BSPDefines::Node::m_faceCount
```

6.35.1.3 m_firstFaceIndex

```
oxyU16 oxygen::BSPDefines::Node::m_firstFaceIndex
```

6.35.1.4 m_maxs

```
oxyS16 oxygen::BSPDefines::Node::m_maxs[3]
```

6.35.1.5 m_mins

```
oxyS16 oxygen::BSPDefines::Node::m_mins[3]
```

6.35.1.6 m_planeIndex

```
oxyU32 oxygen::BSPDefines::Node::m_planeIndex
```

The documentation for this struct was generated from the following file:

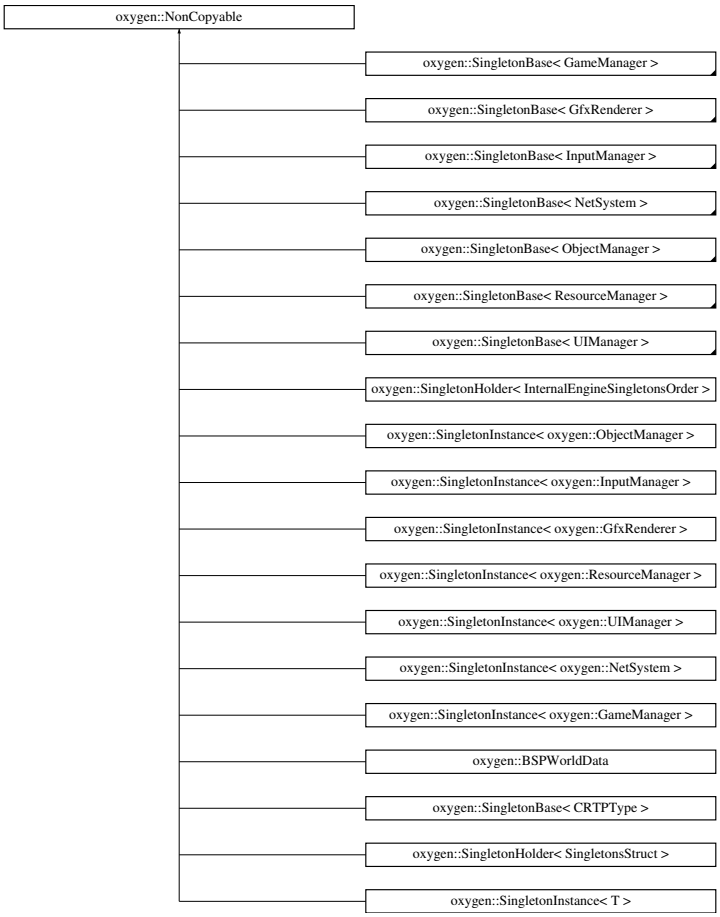
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.36 oxygen::NonCopyable Struct Reference

A non-copyable type to be inherited from. Expresses clearly that a type cannot be copied.

`#include <OxygenTypes.h>`

Inheritance diagram for oxygen::NonCopyable:



Public Member Functions

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

6.36.1 Detailed Description

A non-copyable type to be inherited from. Expresses clearly that a type cannot be copied.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 NonCopyable() [1/2]

`oxygen::NonCopyable::NonCopyable () [default]`

6.36.2.2 NonCopyable() [2/2]

```
oxygen::NonCopyable::NonCopyable (
            const NonCopyable & )  [delete]
```

6.36.3 Member Function Documentation

6.36.3.1 operator=()

```
NonCopyable & oxygen::NonCopyable::operator= (
            const NonCopyable & )  [delete]
```

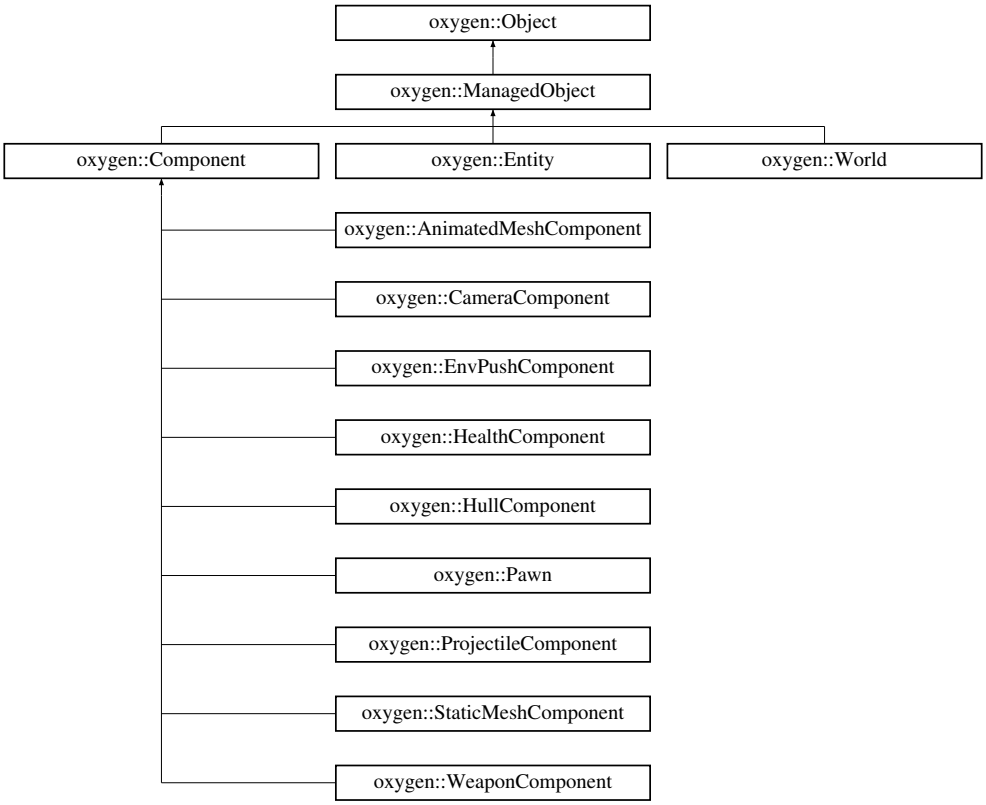
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/OxygenTypes.h

6.37 oxygen::Object Struct Reference

```
#include <Object.h>
```

Inheritance diagram for oxygen::Object:



Public Types

- using SelfType = Object
- using Super = Object

Public Member Functions

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Static Public Member Functions

- static auto GetStaticDescription () -> const ObjectDescription &

6.37.1 Member Typedef Documentation

6.37.1.1 SelfType

```
using oxygen::Object::SelfType = Object
```

6.37.1.2 Super

```
using oxygen::Object::Super = Object
```

6.37.2 Constructor & Destructor Documentation

6.37.2.1 Object()

```
oxygen::Object::Object ()  [default]
```

6.37.2.2 ~Object()

```
virtual oxygen::Object::~Object ()  [virtual], [default]
```

6.37.3 Member Function Documentation

6.37.3.1 Cast() [1/2]

```
template<typename T>
requires std::is_base_of_v<Object, T>
auto oxygen::Object::Cast () -> T*   [inline]
```

6.37.3.2 Cast() [2/2]

```
template<typename T>
requires std::is_base_of_v<Object, T>
auto oxygen::Object::Cast () const -> const T*   [inline]
```

6.37.3.3 GetDescription()

```
virtual auto oxygen::Object::GetDescription () const -> const ObjectDescription&   [inline],
[virtual]
```

6.37.3.4 GetStaticDescription()

```
static auto oxygen::Object::GetStaticDescription () -> const ObjectDescription&   [inline],
[static]
```

6.37.3.5 IsA() [1/2]

```
template<typename T>
requires std::is_base_of_v<Object, T>
auto oxygen::Object::IsA () const -> bool   [inline]
```

6.37.3.6 IsA() [2/2]

```
auto oxygen::Object::IsA (
            const ObjectDescription & desc) const -> bool   [inline]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/Object.h

6.38 oxygen::ObjectDescription Struct Reference

```
#include <Object.h>
```

Public Types

- using constructor_t = Object* (*)(void* p)

Public Attributes

- const ObjectDescription * m_parent {}
- std::string_view m_name {}
- oxyU64 m_id {}
- oxySize m_size {}
- oxySize m_align {}
- constructor_t m_constructor {}

6.38.1 Member Typedef Documentation

6.38.1.1 constructor_t

```
using oxygen::ObjectDescription::constructor_t = Object* (*)(void* p)
```

6.38.2 Member Data Documentation

6.38.2.1 m_align

```
oxySize oxygen::ObjectDescription::m_align {}
```

6.38.2.2 m_constructor

```
constructor_t oxygen::ObjectDescription::m_constructor {}
```

6.38.2.3 m_id

```
oxyU64 oxygen::ObjectDescription::m_id {}
```

6.38.2.4 m_name

```
std::string_view oxygen::ObjectDescription::m_name {}
```

6.38.2.5 m_parent

```
const ObjectDescription* oxygen::ObjectDescription::m_parent {}
```

6.38.2.6 m_size

```
oxySize oxygen::ObjectDescription::m_size {}
```

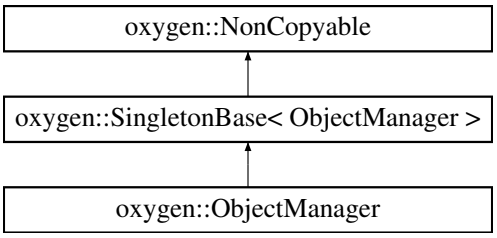
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/Object.h

6.39 oxygen::ObjectManager Struct Reference

`#include <ObjectManager.h>`

Inheritance diagram for oxygen::ObjectManager:



Public Member Functions

- ~ObjectManager ()
- template<typename T>
  requires std::is_base_of_v<Object, T> && std::is_same_v<typename T::SelfType, T>
  auto NewObject (oxyObjectID id=0) -> T *
- auto NewObject (const ObjectDescription &desc, oxyObjectID id=0) -> Object *
- auto DeleteObject (Object *obj, oxyObjectID id=0) -> void
- template<typename T>
  requires std::is_base_of_v<ManagedObject, T> && std::is_same_v<typename T::SelfType, T>
  auto CreateManagedObject (oxyObjectID id=0) -> std::shared_ptr< T >
- auto CreateManagedObject (const ObjectDescription &desc, oxyObjectID id=0) -> std::shared_ptr< ManagedObject >
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto GetObjectPtr (oxyObjectID id) const -> T *
- auto GetObjectPtr (oxyObjectID id) const -> Object *
- auto GetObjectID (Object *obj) const -> oxyObjectID
- template<typename T>
  requires std::is_base_of_v<ManagedObject, T>
  auto GetManagedRef (oxyObjectID id) const -> std::shared_ptr< T >
- auto GetManagedRef (oxyObjectID id) const -> std::shared_ptr< ManagedObject >

Public Member Functions inherited from oxygen::SingletonBase< ObjectManager >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< ObjectManager >

- static auto GetInstance () -> ObjectManager &

6.39.1 Constructor & Destructor Documentation

6.39.1.1 ~ObjectManager()

```
oxygen::ObjectManager::~ObjectManager ()
```

6.39.2 Member Function Documentation

6.39.2.1 CreateManagedObject() [1/2]

```
auto oxygen::ObjectManager::CreateManagedObject (
            const ObjectDescription & desc,
            oxyObjectID id = 0) -> std::shared_ptr<ManagedObject>
```

6.39.2.2 CreateManagedObject() [2/2]

```
template<typename T>
requires std::is_base_of_v<ManagedObject, T> && std::is_same_v<typename T::SelfType, T>
auto oxygen::ObjectManager::CreateManagedObject (
            oxyObjectID id = 0) -> std::shared_ptr<T>
```

6.39.2.3 DeleteObject()

```
auto oxygen::ObjectManager::DeleteObject (
            Object * obj,
            oxyObjectID id = 0) -> void
```

6.39.2.4 GetManagedRef() [1/2]

```
auto oxygen::ObjectManager::GetManagedRef (
            oxyObjectID id) const -> std::shared_ptr<ManagedObject>
```

6.39.2.5 GetManagedRef() [2/2]

```
template<typename T>
requires std::is_base_of_v<ManagedObject, T>
auto oxygen::ObjectManager::GetManagedRef (
            oxyObjectID id) const -> std::shared_ptr<T>
```

6.39.2.6 GetObjectID()

```
auto oxygen::ObjectManager::GetObjectID (
            Object * obj) const -> oxyObjectID
```

6.39.2.7 GetObjectPtr() [1/2]

```
auto oxygen::ObjectManager::GetObjectPtr (
            oxyObjectID id) const -> Object*
```

6.39.2.8 GetObjectPtr() [2/2]

```
template<typename T>
requires std::is_base_of_v<Object, T>
auto oxygen::ObjectManager::GetObjectPtr (
            oxyObjectID id) const -> T*
```

6.39.2.9 NewObject() [1/2]

```
auto oxygen::ObjectManager::NewObject (
            const ObjectDescription & desc,
            oxyObjectID id = 0) -> Object*
```

6.39.2.10 NewObject() [2/2]

```
template<typename T>
requires std::is_base_of_v<Object, T> && std::is_same_v<typename T::SelfType, T>
auto oxygen::ObjectManager::NewObject (
            oxyObjectID id = 0) -> T*
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/ObjectManager.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/ObjectManager.cc

6.40 oxyMat4x4 Struct Reference

```
#include <Defs.h>
```

Public Member Functions

- constexpr auto Determinant () const -> oxyF32
- constexpr auto Transposed () const -> oxyMat4x4
- constexpr auto Transpose () -> oxyMat4x4 &
- constexpr auto operator[ ] (int i) -> oxyF32 *
- constexpr auto operator[ ] (int i) const -> const oxyF32 *
- constexpr auto operator*= (const oxyMat4x4 &other) -> oxyMat4x4 &

Static Public Member Functions

- static constexpr auto Identity () -> oxyMat4x4

Public Attributes

- oxyF32 m [4][4]

6.40.1 Member Function Documentation

6.40.1.1 Determinant()

```
auto oxyMat4x4::Determinant () const -> oxyF32   [inline], [constexpr]
```

6.40.1.2 Identity()

```
static constexpr auto oxyMat4x4::Identity () -> oxyMat4x4   [inline], [static], [constexpr]
```

6.40.1.3 operator*=()

```
auto oxyMat4x4::operator*= (
            const oxyMat4x4 & other) -> oxyMat4x4&   [inline], [constexpr]
```

6.40.1.4 operator[]() [1/2]

```
auto oxyMat4x4::operator[] (
            int i) -> oxyF32*   [inline], [constexpr]
```

6.40.1.5 operator[]() [2/2]

```
auto oxyMat4x4::operator[] (
            int i) const -> const oxyF32*   [inline], [constexpr]
```

6.40.1.6 Transpose()

```
auto oxyMat4x4::Transpose () -> oxyMat4x4&   [inline], [constexpr]
```

6.40.1.7 Transposed()

```
auto oxyMat4x4::Transposed () const -> oxyMat4x4   [inline], [constexpr]
```

6.40.2 Member Data Documentation

6.40.2.1 m

```
oxyF32 oxyMat4x4::m[4][4]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/Defs.h

6.41 oxyQuat Struct Reference

```
#include <Defs.h>
```

Public Member Functions

- constexpr auto MagnitudeSquared () const -> oxyF32
- auto Magnitude () const -> oxyF32
- auto Normalized () const -> oxyQuat
- auto Normalize () -> oxyQuat &
- constexpr auto DotProduct (const oxyQuat &other) const -> oxyF32
- constexpr auto Conjugate () const -> oxyQuat
- auto Inversed () const -> oxyQuat
- auto Inverse () -> oxyQuat &
- constexpr auto operator*= (const oxyQuat &other) -> oxyQuat &

Public Attributes

- oxyF32 x {0.0f}
- oxyF32 y {0.0f}
- oxyF32 z {0.0f}
- oxyF32 w {1.0f}

6.41.1 Member Function Documentation

6.41.1.1 Conjugate()

```
auto oxyQuat::Conjugate () const -> oxyQuat   [inline], [constexpr]
```

6.41.1.2 DotProduct()

```
auto oxyQuat::DotProduct (
            const oxyQuat & other) const -> oxyF32   [inline], [constexpr]
```

6.41.1.3 Inverse()

```
auto oxyQuat::Inverse () -> oxyQuat&   [inline]
```

6.41.1.4 Inversed()

```
auto oxyQuat::Inversed () const -> oxyQuat   [inline]
```

6.41.1.5 Magnitude()

```
auto oxyQuat::Magnitude () const -> oxyF32   [inline]
```

6.41.1.6 MagnitudeSquared()

```
auto oxyQuat::MagnitudeSquared () const -> oxyF32   [inline], [constexpr]
```

6.41.1.7 Normalize()

```
auto oxyQuat::Normalize () -> oxyQuat&   [inline]
```

6.41.1.8 Normalized()

```
auto oxyQuat::Normalized () const -> oxyQuat   [inline]
```

6.41.1.9 operator*=()

```
auto oxyQuat::operator*= (
            const oxyQuat & other) -> oxyQuat&   [inline], [constexpr]
```

6.41.2 Member Data Documentation

6.41.2.1 w

```
oxyF32 oxyQuat::w {1.0f}
```

6.41.2.2 x

```
oxyF32 oxyQuat::x {0.0f}
```

6.41.2.3 y

```
oxyF32 oxyQuat::y {0.0f}
```

6.41.2.4 z

```
oxyF32 oxyQuat::z {0.0f}
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/Defs.h

6.42 oxyVec2 Struct Reference

```
#include <Defs.h>
```

Public Member Functions

- constexpr oxyVec2 ()
- constexpr oxyVec2 (oxyF32 x, oxyF32 y)
- constexpr auto MagnitudeSquared () const -> oxyF32
- auto Magnitude () const -> oxyF32
- auto Normalized () const -> oxyVec2
- auto Normalize () -> oxyVec2 &
- constexpr auto DotProduct (const oxyVec2 &other) const -> oxyF32
- constexpr auto CrossProduct (const oxyVec2 &other) const -> oxyF32
- constexpr auto Conjugate () const -> oxyVec2
- constexpr auto Inversed () const -> oxyVec2
- constexpr auto Inverse () -> oxyVec2 &
- constexpr auto operator+= (const oxyVec2 &other) -> oxyVec2 &
- constexpr auto operator-= (const oxyVec2 &other) -> oxyVec2 &
- constexpr auto operator*= (const oxyVec2 &other) -> oxyVec2 &
- constexpr auto operator/= (const oxyVec2 &other) -> oxyVec2 &
- constexpr auto operator*= (oxyF32 other) -> oxyVec2 &
- constexpr auto operator/= (oxyF32 other) -> oxyVec2 &

Public Attributes

- oxyF32 x
- oxyF32 y

6.42.1 Constructor & Destructor Documentation

6.42.1.1 oxyVec2() [1/2]

```
oxyVec2::oxyVec2 ()  [inline], [constexpr]
```

6.42.1.2 oxyVec2() [2/2]

```
oxyVec2::oxyVec2 (
            oxyF32 x,
            oxyF32 y)  [inline], [constexpr]
```

6.42.2 Member Function Documentation

6.42.2.1 Conjugate()

```
auto oxyVec2::Conjugate () const -> oxyVec2   [inline], [constexpr]
```

6.42.2.2 CrossProduct()

```
auto oxyVec2::CrossProduct (
            const oxyVec2 & other) const -> oxyF32   [inline], [constexpr]
```

6.42.2.3 DotProduct()

```
auto oxyVec2::DotProduct (
            const oxyVec2 & other) const -> oxyF32   [inline], [constexpr]
```

6.42.2.4 Inverse()

```
auto oxyVec2::Inverse () -> oxyVec2&   [inline], [constexpr]
```

6.42.2.5 Inversed()

```
auto oxyVec2::Inversed () const -> oxyVec2   [inline], [constexpr]
```

6.42.2.6 Magnitude()

```
auto oxyVec2::Magnitude () const -> oxyF32   [inline]
```

6.42.2.7 MagnitudeSquared()

```
auto oxyVec2::MagnitudeSquared () const -> oxyF32   [inline], [constexpr]
```

6.42.2.8 Normalize()

```
auto oxyVec2::Normalize () -> oxyVec2&   [inline]
```

6.42.2.9 Normalized()

```
auto oxyVec2::Normalized () const -> oxyVec2   [inline]
```

6.42.2.10 operator*=() [1/2]

```
auto oxyVec2::operator*= (
            const oxyVec2 & other) -> oxyVec2&   [inline], [constexpr]
```

6.42.2.11 operator*=() [2/2]

```
auto oxyVec2::operator*= (
            oxyF32 other) -> oxyVec2&   [inline], [constexpr]
```

6.42.2.12 operator+=()

```
auto oxyVec2::operator+= (
            const oxyVec2 & other) -> oxyVec2&   [inline], [constexpr]
```

6.42.2.13 operator-=()

```
auto oxyVec2::operator-= (
            const oxyVec2 & other) -> oxyVec2&   [inline], [constexpr]
```

6.42.2.14 operator/=() [1/2]

```
auto oxyVec2::operator/= (
            const oxyVec2 & other) -> oxyVec2&   [inline], [constexpr]
```

6.42.2.15 operator/=() [2/2]

```
auto oxyVec2::operator/= (
            oxyF32 other) -> oxyVec2&   [inline], [constexpr]
```

6.42.3 Member Data Documentation

6.42.3.1 x

```
oxyF32 oxyVec2::x
```

6.42.3.2 y

```
oxyF32 oxyVec2::y
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/Defs.h

6.43 oxyVec3 Struct Reference

`#include <Defs.h>`

Public Member Functions

- constexpr oxyVec3 ()
- constexpr oxyVec3 (oxyF32 x, oxyF32 y, oxyF32 z)
- constexpr oxyVec3 (const oxyVec2 &v, oxyF32 z)
- operator oxyVec2 () const
- constexpr auto MagnitudeSquared () const -> oxyF32
- auto Magnitude () const -> oxyF32
- auto Normalized () const -> oxyVec3
- auto Normalize () -> oxyVec3 &
- constexpr auto DotProduct (const oxyVec3 &other) const -> oxyF32
- constexpr auto CrossProduct (const oxyVec3 &other) const -> oxyVec3
- constexpr auto Conjugate () const -> oxyVec3
- constexpr auto Inversed () const -> oxyVec3
- constexpr auto Inverse () -> oxyVec3 &
- constexpr auto operator+= (const oxyVec3 &other) -> oxyVec3 &
- constexpr auto operator-= (const oxyVec3 &other) -> oxyVec3 &
- constexpr auto operator*= (const oxyVec3 &other) -> oxyVec3 &
- constexpr auto operator/= (const oxyVec3 &other) -> oxyVec3 &
- constexpr auto operator*= (oxyF32 other) -> oxyVec3 &
- constexpr auto operator/= (oxyF32 other) -> oxyVec3 &

Public Attributes

- oxyF32 x
- oxyF32 y
- oxyF32 z

6.43.1 Constructor & Destructor Documentation

6.43.1.1 oxyVec3() [1/3]

```
oxyVec3::oxyVec3 ()  [inline], [constexpr]
```

6.43.1.2 oxyVec3() [2/3]

```
oxyVec3::oxyVec3 (
            oxyF32 x,
            oxyF32 y,
            oxyF32 z)  [inline], [constexpr]
```

6.43.1.3 oxyVec3() [3/3]

```
oxyVec3::oxyVec3 (
            const oxyVec2 & v,
            oxyF32 z)  [inline], [constexpr]
```

6.43.2 Member Function Documentation

6.43.2.1 Conjugate()

```
auto oxyVec3::Conjugate () const -> oxyVec3   [inline], [constexpr]
```

6.43.2.2 CrossProduct()

```
auto oxyVec3::CrossProduct (
            const oxyVec3 & other) const -> oxyVec3   [inline], [constexpr]
```

6.43.2.3 DotProduct()

```
auto oxyVec3::DotProduct (
            const oxyVec3 & other) const -> oxyF32   [inline], [constexpr]
```

6.43.2.4 Inverse()

```
auto oxyVec3::Inverse () -> oxyVec3&   [inline], [constexpr]
```

6.43.2.5 Inversed()

```
auto oxyVec3::Inversed () const -> oxyVec3   [inline], [constexpr]
```

6.43.2.6 Magnitude()

```
auto oxyVec3::Magnitude () const -> oxyF32   [inline]
```

6.43.2.7 MagnitudeSquared()

```
auto oxyVec3::MagnitudeSquared () const -> oxyF32   [inline], [constexpr]
```

6.43.2.8 Normalize()

```
auto oxyVec3::Normalize () -> oxyVec3&   [inline]
```

6.43.2.9 Normalized()

```
auto oxyVec3::Normalized () const -> oxyVec3   [inline]
```

6.43.2.10 operator oxyVec2()

```
oxyVec3::operator oxyVec2 () const  [inline]
```

6.43.2.11 operator*=() [1/2]

```
auto oxyVec3::operator*= (
            const oxyVec3 & other) -> oxyVec3&   [inline], [constexpr]
```

6.43.2.12 operator*=() [2/2]

```
auto oxyVec3::operator*= (
            oxyF32 other) -> oxyVec3&   [inline], [constexpr]
```

6.43.2.13 operator+=()

```
auto oxyVec3::operator+= (
            const oxyVec3 & other) -> oxyVec3&   [inline], [constexpr]
```

6.43.2.14 operator-=()

```
auto oxyVec3::operator-= (
            const oxyVec3 & other) -> oxyVec3&   [inline], [constexpr]
```

6.43.2.15 operator/=() [1/2]

```
auto oxyVec3::operator/= (
            const oxyVec3 & other) -> oxyVec3&   [inline], [constexpr]
```

6.43.2.16 operator/=() [2/2]

```
auto oxyVec3::operator/= (
            oxyF32 other) -> oxyVec3&   [inline], [constexpr]
```

6.43.3 Member Data Documentation

6.43.3.1 x

```
oxyF32 oxyVec3::x
```

6.43.3.2 y

```
oxyF32 oxyVec3::y
```

6.43.3.3 z

```
oxyF32 oxyVec3::z
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/Defs.h

6.44 oxyVec4 Struct Reference

```
#include <Defs.h>
```

Public Member Functions

- constexpr oxyVec4 ()
- constexpr oxyVec4 (oxyF32 x, oxyF32 y, oxyF32 z, oxyF32 w)
- constexpr oxyVec4 (const oxyVec3 &v, oxyF32 w)
- constexpr oxyVec4 (const oxyVec2 &v, oxyF32 z, oxyF32 w)
- operator oxyVec3 () const
- operator oxyVec2 () const
- constexpr auto MagnitudeSquared () const -> oxyF32
- auto Magnitude () const -> oxyF32
- auto Normalized () const -> oxyVec4
- auto Normalize () -> oxyVec4 &
- constexpr auto DotProduct (const oxyVec4 &other) const -> oxyF32
- constexpr auto Conjugate () const -> oxyVec4
- constexpr auto Inversed () const -> oxyVec4
- constexpr auto Inverse () -> oxyVec4 &
- constexpr auto CrossProduct (const oxyVec4 &other) const -> oxyVec4
- constexpr auto operator+= (const oxyVec4 &other) -> oxyVec4 &
- constexpr auto operator-= (const oxyVec4 &other) -> oxyVec4 &
- constexpr auto operator*= (const oxyVec4 &other) -> oxyVec4 &
- constexpr auto operator/= (const oxyVec4 &other) -> oxyVec4 &
- constexpr auto operator*= (oxyF32 other) -> oxyVec4 &
- constexpr auto operator/= (oxyF32 other) -> oxyVec4 &

Public Attributes

- oxyF32 x
- oxyF32 y
- oxyF32 z
- oxyF32 w

6.44.1 Constructor & Destructor Documentation

6.44.1.1 oxyVec4() [1/4]

```
oxyVec4::oxyVec4 ()  [inline], [constexpr]
```

6.44.1.2 oxyVec4() [2/4]

```
oxyVec4::oxyVec4 (
            oxyF32 x,
            oxyF32 y,
            oxyF32 z,
            oxyF32 w)  [inline], [constexpr]
```

6.44.1.3 oxyVec4() [3/4]

```
oxyVec4::oxyVec4 (
            const oxyVec3 & v,
            oxyF32 w)  [inline], [constexpr]
```

6.44.1.4 oxyVec4() [4/4]

```
oxyVec4::oxyVec4 (
            const oxyVec2 & v,
            oxyF32 z,
            oxyF32 w)  [inline], [constexpr]
```

6.44.2 Member Function Documentation

6.44.2.1 Conjugate()

```
auto oxyVec4::Conjugate () const -> oxyVec4   [inline], [constexpr]
```

6.44.2.2 CrossProduct()

```
auto oxyVec4::CrossProduct (
            const oxyVec4 & other) const -> oxyVec4   [inline], [constexpr]
```

6.44.2.3 DotProduct()

```
auto oxyVec4::DotProduct (
            const oxyVec4 & other) const -> oxyF32   [inline], [constexpr]
```

6.44.2.4 Inverse()

```
auto oxyVec4::Inverse () -> oxyVec4&   [inline], [constexpr]
```

6.44.2.5 Inversed()

```
auto oxyVec4::Inversed () const -> oxyVec4   [inline], [constexpr]
```

6.44.2.6 Magnitude()

```
auto oxyVec4::Magnitude () const -> oxyF32   [inline]
```

6.44.2.7 MagnitudeSquared()

```
auto oxyVec4::MagnitudeSquared () const -> oxyF32   [inline], [constexpr]
```

6.44.2.8 Normalize()

```
auto oxyVec4::Normalize () -> oxyVec4&   [inline]
```

6.44.2.9 Normalized()

```
auto oxyVec4::Normalized () const -> oxyVec4   [inline]
```

6.44.2.10 operator oxyVec2()

```
oxyVec4::operator oxyVec2 () const  [inline]
```

6.44.2.11 operator oxyVec3()

```
oxyVec4::operator oxyVec3 () const  [inline]
```

6.44.2.12 operator*=() [1/2]

```
auto oxyVec4::operator*= (
            const oxyVec4 & other) -> oxyVec4&   [inline], [constexpr]
```

6.44.2.13 operator*=() [2/2]

```
auto oxyVec4::operator*= (
            oxyF32 other) -> oxyVec4&   [inline], [constexpr]
```

6.44.2.14 operator+=()

```
auto oxyVec4::operator+= (
            const oxyVec4 & other) -> oxyVec4&   [inline], [constexpr]
```

6.44.2.15 operator-=()

```
auto oxyVec4::operator-= (
            const oxyVec4 & other) -> oxyVec4&   [inline], [constexpr]
```

6.44.2.16 operator/=() [1/2]

```
auto oxyVec4::operator/= (
            const oxyVec4 & other) -> oxyVec4&   [inline], [constexpr]
```

6.44.2.17 operator/=() [2/2]

```
auto oxyVec4::operator/= (
            oxyF32 other) -> oxyVec4&   [inline], [constexpr]
```

6.44.3 Member Data Documentation

6.44.3.1 w

```
oxyF32 oxyVec4::w
```

6.44.3.2 x

```
oxyF32 oxyVec4::x
```

6.44.3.3 y

```
oxyF32 oxyVec4::y
```

6.44.3.4 z

```
oxyF32 oxyVec4::z
```

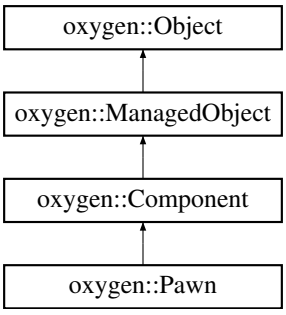
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/Defs.h

6.45 oxygen::Pawn Struct Reference

```
#include <Pawn.h>
```

Inheritance diagram for oxygen::Pawn:



Public Member Functions

- OXYGENOBJECT (Pawn, Component)
- auto GetState () const -> PawnState
- auto GetStance () const -> PawnStance
- auto GetEquippedWeapon () const -> const std::shared_ptr< struct WeaponComponent > &
- auto GetEquippedRightHandedWeapon () const -> const std::shared_ptr< struct WeaponComponent > &

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (oxyF32 deltaTimeSeconds) -> void override
- auto Render () const -> void override

Friends

- struct GameManager
- struct WeaponComponent

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.45.1 Member Function Documentation

6.45.1.1 GetEquippedRightHandedWeapon()

```
auto oxygen::Pawn::GetEquippedRightHandedWeapon () const -> const std::shared_ptr<struct
WeaponComponent>&   [inline]
```

6.45.1.2 GetEquippedWeapon()

```
auto oxygen::Pawn::GetEquippedWeapon () const -> const std::shared_ptr<struct WeaponComponent>&
[inline]
```

6.45.1.3 GetStance()

```
auto oxygen::Pawn::GetStance () const -> PawnStance   [inline]
```

6.45.1.4 GetState()

```
auto oxygen::Pawn::GetState () const -> PawnState   [inline]
```

6.45.1.5 OXYGENOBJECT()

```
oxygen::Pawn::OXYGENOBJECT (
            Pawn ,
            Component )
```

6.45.1.6 Render()

```
auto oxygen::Pawn::Render () const -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

6.45.1.7 Update()

```
auto oxygen::Pawn::Update (
            oxyF32 deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

6.45.2 Friends And Related Symbol Documentation

6.45.2.1 GameManager

```
friend struct GameManager  [friend]
```

6.45.2.2 WeaponComponent

```
friend struct WeaponComponent  [friend]
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/Pawn/Pawn.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/Pawn/Pawn.cc

6.46 oxygen::BSPDefines::Plane Struct Reference

```
#include <BSP.h>
```

Public Attributes

- oxyF32 m_normal [3]
- oxyF32 m_dist
- oxyU32 m_type

6.46.1 Member Data Documentation

6.46.1.1 m_dist

```
oxyF32 oxygen::BSPDefines::Plane::m_dist
```

6.46.1.2 m_normal

```
oxyF32 oxygen::BSPDefines::Plane::m_normal[3]
```

6.46.1.3 m_type

```
oxyU32 oxygen::BSPDefines::Plane::m_type
```

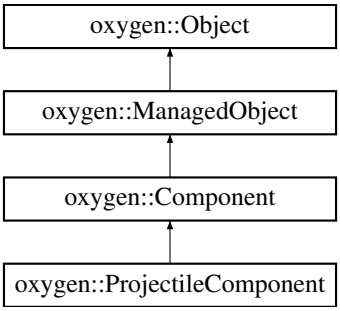
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.47 oxygen::ProjectileComponent Struct Reference

```
#include <ProjectileComponent.h>
```

Inheritance diagram for oxygen::ProjectileComponent:



Public Member Functions

- OXYGENOBJECT (ProjectileComponent, Component)
- auto SetBouncesLeft (oxyS32 bounces) -> void
- auto SetDamage (oxyF32 damage) -> void
- auto SetDamageRadius (oxyF32 radius) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (oxyF32 deltaTimeSeconds) -> void override

Protected Member Functions inherited from oxygen::Component

- virtual auto Render () const -> void

Friends

- struct GameManager

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.47.1 Member Function Documentation

6.47.1.1 OXYGENOBJECT()

```
oxygen::ProjectileComponent::OXYGENOBJECT (
            ProjectileComponent ,
            Component )
```

6.47.1.2 SetBouncesLeft()

```
auto oxygen::ProjectileComponent::SetBouncesLeft (
            oxyS32 bounces) -> void   [inline]
```

6.47.1.3 SetDamage()

```
auto oxygen::ProjectileComponent::SetDamage (
            oxyF32 damage) -> void   [inline]
```

6.47.1.4 SetDamageRadius()

```
auto oxygen::ProjectileComponent::SetDamageRadius (
            oxyF32 radius) -> void   [inline]
```

6.47.1.5 Update()

```
auto oxygen::ProjectileComponent::Update (
            oxyF32 deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

6.47.2 Friends And Related Symbol Documentation

6.47.2.1 GameManager

```
friend struct GameManager  [friend]
```

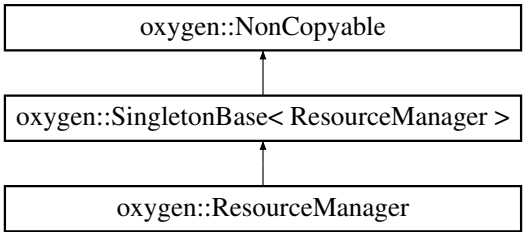
The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/ProjectileComponent/ProjectileComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/ProjectileComponent/ProjectileComponent.cc

6.48 oxygen::ResourceManager Struct Reference

```
#include <ResourceManager.h>
```

Inheritance diagram for oxygen::ResourceManager:



Public Member Functions

- auto LoadStaticMesh (std::string_view name) -> std::shared_ptr< const StaticMeshResource >
- auto LoadAnimatedMesh (std::string_view name) -> std::shared_ptr< const AnimatedMeshResource >

Public Member Functions inherited from oxygen::SingletonBase< ResourceManager >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< ResourceManager >

- static auto GetInstance () -> ResourceManager &

6.48.1 Member Function Documentation

6.48.1.1 LoadAnimatedMesh()

```
auto oxygen::ResourceManager::LoadAnimatedMesh (
            std::string_view name) -> std::shared_ptr<const AnimatedMeshResource>
```

6.48.1.2 LoadStaticMesh()

```
auto oxygen::ResourceManager::LoadStaticMesh (
            std::string_view name) -> std::shared_ptr<const StaticMeshResource>
```

The documentation for this struct was generated from the following files:

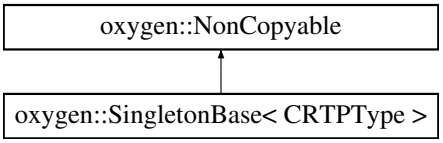
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/ResourceManager.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/ResourceManager.cc

6.49 oxygen::SingletonBase< CRTPType > Struct Template Reference

Singleton base class. All singletons should inherit from this class. Uses the curiously recurring template pattern and provides the GetInstance() static method to access the singleton.

`#include <Singleton.h>`

Inheritance diagram for oxygen::SingletonBase< CRTPType >:



Public Member Functions

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Static Public Member Functions

- static auto GetInstance () -> CRTPType &

6.49.1 Detailed Description

template<typename CRTPType>
struct oxygen::SingletonBase< CRTPType >

Singleton base class. All singletons should inherit from this class. Uses the curiously recurring template pattern and provides the GetInstance() static method to access the singleton.

Template Parameters

CRTPType	The type of the singleton.

6.49.2 Constructor & Destructor Documentation

6.49.2.1 SingletonBase()

```
template<typename CRTPType>
oxygen::SingletonBase< CRTPType >::SingletonBase ()  [inline]
```

6.49.2.2 ~SingletonBase()

```
template<typename CRTPType>
oxygen::SingletonBase< CRTPType >::~SingletonBase ()  [inline]
```

6.49.3 Member Function Documentation

6.49.3.1 GetInstance()

```
template<typename CRTPType>
static auto oxygen::SingletonBase< CRTPType >::GetInstance () -> CRTPType&   [inline], [static]
```

The documentation for this struct was generated from the following file:

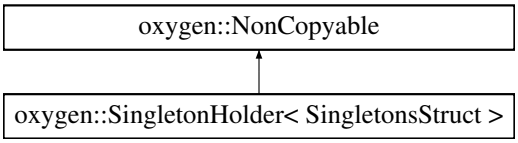
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Singleton/Singleton.h

6.50 oxygen::SingletonHolder< SingletonsStruct > Struct Template Reference

Storage of a group of singletons. Used to explicitly construct and destruct singletons.

`#include <Singleton.h>`

Inheritance diagram for oxygen::SingletonHolder< SingletonsStruct >:



Static Public Member Functions

- static auto Construct () -> void
- static auto Destruct () -> void

Additional Inherited Members

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

6.50.1 Detailed Description

template<typename SingletonsStruct>
struct oxygen::SingletonHolder< SingletonsStruct >

Storage of a group of singletons. Used to explicitly construct and destruct singletons.

Template Parameters

SingletonsStruct	A struct containing SingletonInstance objects.

6.50.2 Member Function Documentation

6.50.2.1 Construct()

```
template<typename SingletonsStruct>
static auto oxygen::SingletonHolder< SingletonsStruct >::Construct () -> void   [inline],
[static]
```

6.50.2.2 Destruct()

```
template<typename SingletonsStruct>
static auto oxygen::SingletonHolder< SingletonsStruct >::Destruct () -> void   [inline],
[static]
```

The documentation for this struct was generated from the following file:

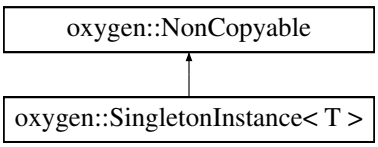
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Singleton/Singleton.h

6.51 oxygen::SingletonInstance< T > Struct Template Reference

Singleton instance, should only be in an object templated to SingletonHolder. Contains the actual storage buffer for the type T.

```
#include <Singleton.h>
```

Inheritance diagram for oxygen::SingletonInstance< T >:



Public Member Functions

- template<typename... Args>
  SingletonInstance (Args &&... args)
- ~SingletonInstance ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Static Public Member Functions

- static auto GetInstance () -> T &

6.51.1 Detailed Description

template<typename T>
struct oxygen::SingletonInstance< T >

Singleton instance, should only be in an object templated to SingletonHolder. Contains the actual storage buffer for the type T.

Template Parameters

T	The type of the singleton.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 SingletonInstance()

```
template<typename T>
template<typename...  Args>
oxygen::SingletonInstance< T >::SingletonInstance (
            Args &&...  args)  [inline]
```

6.51.2.2 ~SingletonInstance()

```
template<typename T>
oxygen::SingletonInstance< T >::~SingletonInstance ()  [inline]
```

6.51.3 Member Function Documentation

6.51.3.1 GetInstance()

```
template<typename T>
static auto oxygen::SingletonInstance< T >::GetInstance () -> T&   [inline], [static]
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Singleton/Singleton.h

6.52 oxygen::SPSCQueue< T, N > Struct Template Reference

```
#include <SPSCQueue.h>
```

Public Member Functions

- SPSCQueue ()
- template<typename T>
  auto TryPush (T &&value) -> bool
- auto TryPop (T &valueOut) -> bool

6.52.1 Constructor & Destructor Documentation

6.52.1.1 SPSCQueue()

```
template<typename T, int N>
oxygen::SPSCQueue< T, N >::SPSCQueue ()  [inline]
```

6.52.2 Member Function Documentation

6.52.2.1 TryPop()

```
template<typename T, int N>
auto oxygen::SPSCQueue< T, N >::TryPop (
            T & valueOut) -> bool   [inline]
```

6.52.2.2 TryPush()

```
template<typename T, int N>
template<typename T>
auto oxygen::SPSCQueue< T, N >::TryPush (
            T && value) -> bool   [inline]
```

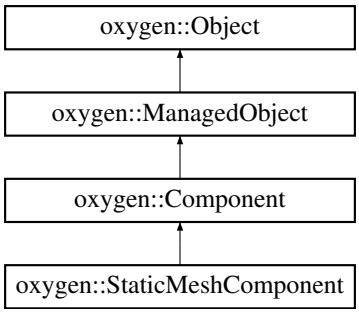
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Containers/SPSCQueue.h

6.53 oxygen::StaticMeshComponent Struct Reference

```
#include <StaticMeshComponent.h>
```

Inheritance diagram for oxygen::StaticMeshComponent:



Public Member Functions

- OXYGENOBJECT (StaticMeshComponent, Component)
- auto LoadByName (std::string_view name) -> oxyBool
- auto SetLocalOffset (const oxyVec3 &offset) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Render () const -> void override

Protected Member Functions inherited from oxygen::Component

- virtual auto Update (oxyF32 deltaTimeSeconds) -> void

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.53.1 Member Function Documentation

6.53.1.1 LoadByName()

```
auto oxygen::StaticMeshComponent::LoadByName (
            std::string_view name) -> oxyBool
```

6.53.1.2 OXYGENOBJECT()

```
oxygen::StaticMeshComponent::OXYGENOBJECT (
            StaticMeshComponent ,
            Component )
```

6.53.1.3 Render()

```
auto oxygen::StaticMeshComponent::Render () const -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

6.53.1.4 SetLocalOffset()

```
auto oxygen::StaticMeshComponent::SetLocalOffset (
            const oxyVec3 & offset) -> void   [inline]
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/StaticMeshComponent/StaticMeshComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/StaticMeshComponent/StaticMeshComponent.cc

6.54 oxygen::StaticMeshPointDef Struct Reference

```
#include <StaticMeshResource.h>
```

Public Attributes

- oxyU32 m_hash
- oxyVec3 m_position

6.54.1 Member Data Documentation

6.54.1.1 m_hash

```
oxyU32 oxygen::StaticMeshPointDef::m_hash
```

6.54.1.2 m_position

```
oxyVec3 oxygen::StaticMeshPointDef::m_position
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/StaticMeshResource.h

6.55 oxygen::StaticMeshResource Struct Reference

```
#include <StaticMeshResource.h>
```

Public Attributes

- std::vector< StaticMeshPointDef > m_points
- std::vector< StaticMeshTri > m_tris
- std::string m_texname

6.55.1 Member Data Documentation

6.55.1.1 m_points

```
std::vector<StaticMeshPointDef> oxygen::StaticMeshResource::m_points
```

6.55.1.2 m_texname

```
std::string oxygen::StaticMeshResource::m_texname
```

6.55.1.3 m_tris

```
std::vector<StaticMeshTri> oxygen::StaticMeshResource::m_tris
```

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/StaticMeshResource.h

6.56 oxygen::StaticMeshTri Struct Reference

`#include <StaticMeshResource.h>`

Public Attributes

- StaticMeshVertex m_vertices [3]

6.56.1 Member Data Documentation

6.56.1.1 m_vertices

`StaticMeshVertex oxygen::StaticMeshTri::m_vertices[3]`

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/StaticMeshResource.h

6.57 oxygen::StaticMeshVertex Struct Reference

`#include <StaticMeshResource.h>`

Public Attributes

- oxyVec3 m_position
- oxyVec2 m_uv

6.57.1 Member Data Documentation

6.57.1.1 m_position

`oxyVec3 oxygen::StaticMeshVertex::m_position`

6.57.1.2 m_uv

`oxyVec2 oxygen::StaticMeshVertex::m_uv`

The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Resources/StaticMeshResource.h

6.58 oxygen::BSPDefines::TexInfo Struct Reference

`#include <BSP.h>`

Public Attributes

- oxyF32 m_vecs [2][4]
- oxyS32 m_mipTexIndex
- oxyS32 m_flags

6.58.1 Member Data Documentation

6.58.1.1 m_flags

`oxyS32 oxygen::BSPDefines::TexInfo::m_flags`

6.58.1.2 m_mipTexIndex

`oxyS32 oxygen::BSPDefines::TexInfo::m_mipTexIndex`

6.58.1.3 m_vecs

`oxyF32 oxygen::BSPDefines::TexInfo::m_vecs[2][4]`

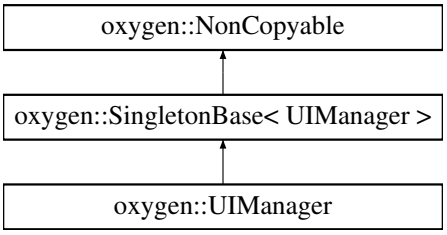
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.59 oxygen::UIManager Struct Reference

`#include <UIManager.h>`

Inheritance diagram for oxygen::UIManager:



Public Member Functions

- auto Render () const -> void
- auto Update () -> void
- auto DisplayPopup (std::string message) -> void

Public Member Functions inherited from oxygen::SingletonBase< UIManager >

- SingletonBase ()
- ~SingletonBase ()

Public Member Functions inherited from oxygen::NonCopyable

- NonCopyable ()=default
- NonCopyable (const NonCopyable &)=delete
- NonCopyable & operator= (const NonCopyable &)=delete

Additional Inherited Members

Static Public Member Functions inherited from oxygen::SingletonBase< UIManager >

- static auto GetInstance () -> UIManager &

6.59.1 Member Function Documentation

6.59.1.1 DisplayPopup()

```
auto oxygen::UIManager::DisplayPopup (
            std::string message) -> void
```

6.59.1.2 Render()

```
auto oxygen::UIManager::Render () const -> void
```

6.59.1.3 Update()

```
auto oxygen::UIManager::Update () -> void
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/UI/UIManager.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/UI/UIManager.cc

6.60 oxygen::BSPDefines::Vertex Struct Reference

`#include <BSP.h>`

Public Attributes

- oxyF32 m_position [3]

6.60.1 Member Data Documentation

6.60.1.1 m_position

`oxyF32 oxygen::BSPDefines::Vertex::m_position[3]`

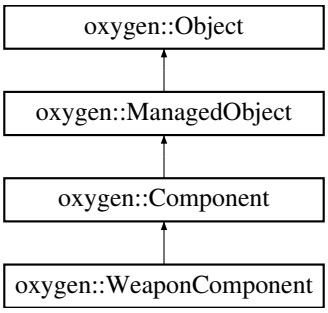
The documentation for this struct was generated from the following file:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h

6.61 oxygen::WeaponComponent Struct Reference

`#include <WeaponComponent.h>`

Inheritance diagram for oxygen::WeaponComponent:



Public Member Functions

- OXYGENOBJECT (WeaponComponent, Component)
- auto HasInfiniteReserve () const -> oxyBool
- auto HasInfiniteClip () const -> oxyBool
- auto GetReserveAmmo () const -> oxyU32
- auto GetClipAmmo () const -> oxyU32
- auto GetMaxClipAmmo () const -> oxyU32
- auto GetMaxReserveAmmo () const -> oxyU32
- auto GetBulletsPerShot () const -> oxyU32
- auto GetRPM () const -> oxyU32
- auto GetTimeToReload () const -> oxyF32
- auto GetSpreadRadians () const -> oxyVec2

- auto GetFireType () const -> WeaponFireType
- auto GetDestroyOnFire () const -> oxyBool
- auto GetRightHanded () const -> oxyBool
- auto GetCanDrop () const -> oxyBool
- auto SetInfiniteReserve (oxyBool infinite) -> void
- auto SetInfiniteClip (oxyBool infinite) -> void
- auto SetReserveAmmo (oxyU32 ammo) -> void
- auto SetClipAmmo (oxyU32 ammo) -> void
- auto SetMaxClipAmmo (oxyU32 ammo) -> void
- auto SetMaxReserveAmmo (oxyU32 ammo) -> void
- auto SetBulletsPerShot (oxyU32 bullets) -> void
- auto SetRPM (oxyU32 rpm) -> void
- auto SetTimeToReload (oxyF32 time) -> void
- auto SetSpreadRadians (oxyVec2 spread) -> void
- auto SetFireType (WeaponFireType type) -> void
- auto SetDestroyOnFire (oxyBool destroy) -> void
- auto SetRightHanded (oxyBool right) -> void
- auto SetCanDrop (oxyBool canDrop) -> void

Public Member Functions inherited from oxygen::Component

- OXYGENOBJECT (Component, ManagedObject)
- auto GetEntity () const -> std::shared_ptr< Entity >
- auto IsEnabled () const -> oxyBool
- auto SetEnabled (oxyBool enabled) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Protected Member Functions

- auto Update (oxyF32 deltaTimeSeconds) -> void override

Protected Member Functions inherited from oxygen::Component

- virtual auto Render () const -> void

Friends

- struct Pawn
- struct GameManager

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.61.1 Member Function Documentation

6.61.1.1 GetBulletsPerShot()

```
auto oxygen::WeaponComponent::GetBulletsPerShot () const -> oxyU32   [inline]
```

6.61.1.2 GetCanDrop()

```
auto oxygen::WeaponComponent::GetCanDrop () const -> oxyBool   [inline]
```

6.61.1.3 GetClipAmmo()

```
auto oxygen::WeaponComponent::GetClipAmmo () const -> oxyU32   [inline]
```

6.61.1.4 GetDestroyOnFire()

```
auto oxygen::WeaponComponent::GetDestroyOnFire () const -> oxyBool   [inline]
```

6.61.1.5 GetFireType()

```
auto oxygen::WeaponComponent::GetFireType () const -> WeaponFireType   [inline]
```

6.61.1.6 GetMaxClipAmmo()

```
auto oxygen::WeaponComponent::GetMaxClipAmmo () const -> oxyU32   [inline]
```

6.61.1.7 GetMaxReserveAmmo()

```
auto oxygen::WeaponComponent::GetMaxReserveAmmo () const -> oxyU32   [inline]
```

6.61.1.8 GetReserveAmmo()

```
auto oxygen::WeaponComponent::GetReserveAmmo () const -> oxyU32   [inline]
```

6.61.1.9 GetRightHanded()

```
auto oxygen::WeaponComponent::GetRightHanded () const -> oxyBool   [inline]
```

6.61.1.10 GetRPM()

```
auto oxygen::WeaponComponent::GetRPM () const -> oxyU32   [inline]
```

6.61.1.11 GetSpreadRadians()

```
auto oxygen::WeaponComponent::GetSpreadRadians () const -> oxyVec2   [inline]
```

6.61.1.12 GetTimeToReload()

```
auto oxygen::WeaponComponent::GetTimeToReload () const -> oxyF32   [inline]
```

6.61.1.13 HasInfiniteClip()

```
auto oxygen::WeaponComponent::HasInfiniteClip () const -> oxyBool   [inline]
```

6.61.1.14 HasInfiniteReserve()

```
auto oxygen::WeaponComponent::HasInfiniteReserve () const -> oxyBool   [inline]
```

6.61.1.15 OXYGENOBJECT()

```
oxygen::WeaponComponent::OXYGENOBJECT (
            WeaponComponent ,
            Component )
```

6.61.1.16 SetBulletsPerShot()

```
auto oxygen::WeaponComponent::SetBulletsPerShot (
            oxyU32 bullets) -> void   [inline]
```

6.61.1.17 SetCanDrop()

```
auto oxygen::WeaponComponent::SetCanDrop (
            oxyBool canDrop) -> void   [inline]
```

6.61.1.18 SetClipAmmo()

```
auto oxygen::WeaponComponent::SetClipAmmo (
            oxyU32 ammo) -> void   [inline]
```

6.61.1.19 SetDestroyOnFire()

```
auto oxygen::WeaponComponent::SetDestroyOnFire (
            oxyBool destroy) -> void   [inline]
```

6.61.1.20 SetFireType()

```
auto oxygen::WeaponComponent::SetFireType (
            WeaponFireType type) -> void   [inline]
```

6.61.1.21 SetInfiniteClip()

```
auto oxygen::WeaponComponent::SetInfiniteClip (
            oxyBool infinite) -> void   [inline]
```

6.61.1.22 SetInfiniteReserve()

```
auto oxygen::WeaponComponent::SetInfiniteReserve (
            oxyBool infinite) -> void   [inline]
```

6.61.1.23 SetMaxClipAmmo()

```
auto oxygen::WeaponComponent::SetMaxClipAmmo (
            oxyU32 ammo) -> void   [inline]
```

6.61.1.24 SetMaxReserveAmmo()

```
auto oxygen::WeaponComponent::SetMaxReserveAmmo (
            oxyU32 ammo) -> void   [inline]
```

6.61.1.25 SetReserveAmmo()

```
auto oxygen::WeaponComponent::SetReserveAmmo (
            oxyU32 ammo) -> void   [inline]
```

6.61.1.26 SetRightHanded()

```
auto oxygen::WeaponComponent::SetRightHanded (
            oxyBool right) -> void   [inline]
```

6.61.1.27 SetRPM()

```
auto oxygen::WeaponComponent::SetRPM (
            oxyU32 rpm) -> void   [inline]
```

6.61.1.28 SetSpreadRadians()

```
auto oxygen::WeaponComponent::SetSpreadRadians (
            oxyVec2 spread) -> void   [inline]
```

6.61.1.29 SetTimeToReload()

```
auto oxygen::WeaponComponent::SetTimeToReload (
            oxyF32 time) -> void   [inline]
```

6.61.1.30 Update()

```
auto oxygen::WeaponComponent::Update (
            oxyF32 deltaTimeSeconds) -> void  [override], [protected], [virtual]
```

Reimplemented from oxygen::Component.

6.61.2 Friends And Related Symbol Documentation

6.61.2.1 GameManager

```
friend struct GameManager  [friend]
```

6.61.2.2 Pawn

```
friend struct Pawn  [friend]
```

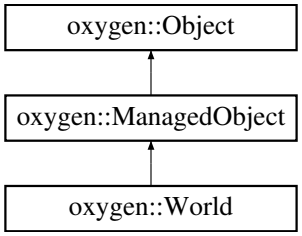
The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/WeaponComponent/WeaponComponent.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/WeaponComponent/WeaponComponent.cc

6.62 oxygen::World Struct Reference

```
#include <World.h>
```

Inheritance diagram for oxygen::World:



Classes

- struct LineTraceResult

Public Member Functions

- OXYGENOBJECT (World, ManagedObject)
- auto GetEntityList () const -> const std::vector< std::shared_ptr< struct Entity > > &
- auto RandomPlayerSpawn () const -> oxyVec3
- auto FindLeaf (const oxyVec3 &position, oxySize modelIndex) const -> const BSPDefines::Leaf *
- auto HullTrace (CollisionHull hull, const oxyVec3 &start, const oxyVec3 &end, LineTraceResult &result) const -> oxyBool
- auto LineTrace (const oxyVec3 &start, const oxyVec3 &end, const struct Entity *self, LineTraceResult &result) const -> oxyBool
- auto CalculateHullSlideMovement (CollisionHull hull, const oxyVec3 &position, const oxyVec3 &distance) -> oxyVec3
- auto SpawnEntity (oxyObjectID id=0) -> std::shared_ptr< Entity >
- auto RemoveEntity (struct Entity *ent) -> void
- auto GetLocalPlayer () const -> std::weak_ptr< Entity >
- auto SetLocalPlayer (std::shared_ptr< Entity > player) -> void

Public Member Functions inherited from oxygen::ManagedObject

- OXYGENOBJECT (ManagedObject, Object)
- auto GetObjectID () const -> oxyObjectID
- template<typename RefType>
  requires std::is_base_of_v<ManagedObject, RefType>
  auto GetHardRef () const -> std::shared_ptr< RefType >

Public Member Functions inherited from oxygen::Object

- Object ()=default
- virtual ~Object ()=default
- virtual auto GetDescription () const -> const ObjectDescription &
- auto IsA (const ObjectDescription &desc) const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto IsA () const -> bool
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () -> T *
- template<typename T>
  requires std::is_base_of_v<Object, T>
  auto Cast () const -> const T *

Public Attributes

- std::unique_ptr< const BSPWorldData > m_bspData {}

Friends

- struct GameManager
- auto LoadWorld (std::string_view name) -> std::shared_ptr< World >

Additional Inherited Members

Public Types inherited from oxygen::Object

- using SelfType = Object
- using Super = Object

Static Public Member Functions inherited from oxygen::Object

- static auto GetStaticDescription () -> const ObjectDescription &

6.62.1 Member Function Documentation

6.62.1.1 CalculateHullSlideMovement()

```
auto oxygen::World::CalculateHullSlideMovement (
            CollisionHull hull,
            const oxyVec3 & position,
            const oxyVec3 & distance) -> oxyVec3
```

6.62.1.2 FindLeaf()

```
auto oxygen::World::FindLeaf (
            const oxyVec3 & position,
            oxySize modelIndex) const -> const BSPDefines::Leaf*
```

6.62.1.3 GetEntityList()

```
auto oxygen::World::GetEntityList () const -> const std::vector<std::shared_ptr<struct Entity>>&
[inline]
```

6.62.1.4 GetLocalPlayer()

```
auto oxygen::World::GetLocalPlayer () const -> std::weak_ptr<Entity>   [inline]
```

6.62.1.5 HullTrace()

```
auto oxygen::World::HullTrace (
            CollisionHull hull,
            const oxyVec3 & start,
            const oxyVec3 & end,
            LineTraceResult & result) const -> oxyBool
```

6.62.1.6 LineTrace()

```
auto oxygen::World::LineTrace (
            const oxyVec3 & start,
            const oxyVec3 & end,
            const struct Entity * self,
            LineTraceResult & result) const -> oxyBool
```

6.62.1.7 OXYGENOBJECT()

```
oxygen::World::OXYGENOBJECT (
            World ,
            ManagedObject )
```

6.62.1.8 RandomPlayerSpawn()

```
auto oxygen::World::RandomPlayerSpawn () const -> oxyVec3
```

6.62.1.9 RemoveEntity()

```
auto oxygen::World::RemoveEntity (
            struct Entity * ent) -> void
```

6.62.1.10 SetLocalPlayer()

```
auto oxygen::World::SetLocalPlayer (
            std::shared_ptr< Entity > player) -> void
```

6.62.1.11 SpawnEntity()

```
auto oxygen::World::SpawnEntity (
            oxyObjectID id = 0) -> std::shared_ptr<Entity>
```

6.62.2 Friends And Related Symbol Documentation

6.62.2.1 GameManager

```
friend struct GameManager  [friend]
```

6.62.2.2 LoadWorld

```
auto LoadWorld (
            std::string_view name) -> std::shared_ptr<World>  [friend]
```

6.62.3 Member Data Documentation

6.62.3.1 m_bspData

```
std::unique_ptr<const BSPWorldData> oxygen::World::m_bspData {}
```

The documentation for this struct was generated from the following files:

- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/World.h
- C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/World.cc

Chapter 7

File Documentation

7.1 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/AnimatedMeshComponent/AnimatedMesh↩ Component.cc File Reference

```
#include <OxygenPCH.h>
#include "AnimatedMeshComponent.h"
#include "Entity/Entity.h"
#include "Resources/ResourceManager.h"
#include "Resources/AnimatedMeshResource.h"
#include "Resources/StaticMeshResource.h"
#include "Gfx/GfxRenderer.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.2 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/AnimatedMeshComponent/AnimatedMeshComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::AnimatedMeshComponent

Namespaces

- namespace oxygen

7.3 AnimatedMeshComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct AnimatedMeshResource;
00008     struct AnimatedMeshComponent final : Component
00009     {
00010         OXYGENOBJECT(AnimatedMeshComponent, Component);
00011
00012         auto LoadByName(std::string_view name) -> oxyBool;
00013
00014         auto BeginAnimation(oxyU32 animHash, oxyBool loop = true) -> void;
00015
00016         auto GetCurrenntAnimationHash() const -> oxyU32
00017         {
00018             return m_animHash;
00019         }
00020
00021         auto SetLocalOffset(const oxyVec3& offset) -> void
00022         {
00023             m_localOffset = offset;
00024         }
00025
00026         auto SetLocalRotation(const oxyQuat& rotation) -> void
00027         {
00028             m_localRotation = rotation;
00029         }
00030
00031       protected:
00032         auto Update(float deltaTimeSeconds) -> void override;
00033         auto Render() const -> void override;
00034
00035       private:
00036         oxyU32 m_animHash{};
00037         oxyF32 m_animTotalTime{};
00038         oxyF32 m_animCurrentFrameTime{};
00039         oxyF32 m_animLerpAlpha{};
00040         oxyBool m_loopAnim{};
00041         oxyVec3 m_localOffset{};
00042         oxyQuat m_localRotation{};
00043         const std::vector<oxyVec3>* m_currentFrame{};
00044         const std::vector<oxyVec3>* m_nextFrame{};
00045         std::shared_ptr<const AnimatedMeshResource> m_resource;
00046         std::shared_ptr<const struct GfxTexture> m_texture;
00047     };
00048 } // namespace oxygen
```

7.4 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/CameraComponent/CameraComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "CameraComponent.h"
#include "Entity/Entity.h"
#include "Gfx/GfxRenderer.h"
```

Namespaces

- namespace oxygen

7.5 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/CameraComponent/CameraComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::CameraComponent

Namespaces

- namespace oxygen

7.6 CameraComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct CameraComponent final : Component
00008     {
00009         OXYGENOBJECT(CameraComponent, Component);
00010
00011         auto GetCameraLocalOffset() const -> const oxyVec3&
00012         {
00013             return m_cameraLocalOffset;
00014         }
00015         auto GetEuler() const -> const oxyVec3&
00016         {
00017             return m_cameraEuler;
00018         }
00019         auto GetCameraForward() const -> const oxyVec3&
00020         {
00021             return m_cameraForward;
00022         }
00023         auto GetCameraUp() const -> const oxyVec3&
00024         {
00025             return m_cameraUp;
00026         }
00027         auto GetCameraRight() const -> const oxyVec3&
00028         {
00029             return m_cameraRight;
00030         }
00031         auto GetVerticalFov() const -> oxyF32
00032         {
00033             return m_verticalFovRad;
00034         }
00035         auto GetNearClip() const -> oxyF32
00036         {
00037             return m_nearClip;
00038         }
00039         auto GetFarClip() const -> oxyF32
00040         {
00041             return m_farClip;
00042         }
00043         auto GetViewMatrix() const -> const oxyMat4x4&
00044         {
00045             return m_viewMatrix;
00046         }
00047         auto GetProjectionMatrix() const -> const oxyMat4x4&
00048         {
00049             return m_projectionMatrix;
00050         }
00051         auto GetViewProjectionMatrix() const -> const oxyMat4x4&
00052         {
```

```
00053             return m_viewProjectionMatrix;
00054         }
00055
00056         auto SetEuler(const oxyVec3& euler) -> void
00057         {
00058             m_cameraEuler = euler;
00059         }
00060         auto SetLocalOffset(const oxyVec3& localOffset) -> void
00061         {
00062             m_cameraLocalOffset = localOffset;
00063         }
00064         auto SetNearClip(oxyF32 nearClip) -> void
00065         {
00066             m_nearClip = nearClip;
00067         }
00068         auto SetFarClip(oxyF32 farClip) -> void
00069         {
00070             m_farClip = farClip;
00071         }
00072
00073       protected:
00074         auto Update(oxyF32 deltaTimeSeconds) -> void override;
00075
00076       private:
00077         oxyVec3 m_cameraLocalOffset{};
00078         oxyVec3 m_cameraEuler{};
00079         oxyVec3 m_cameraForward{1.f, 0.f, 0.f};
00080         oxyVec3 m_cameraUp{0.f, 0.f, 1.f};
00081         oxyVec3 m_cameraRight{0.f, 1.f, 0.f};
00082         oxyF32 m_verticalFovRad{90.0f * Math::k_degToRad};
00083         oxyF32 m_nearClip{4.f};
00084         oxyF32 m_farClip{4000.f};
00085         oxyMat4x4 m_viewMatrix{};
00086         oxyMat4x4 m_projectionMatrix{};
00087         oxyMat4x4 m_viewProjectionMatrix{};
00088
00089         oxyBool m_updateEntityYaw{true};
00090     };
00091 }; // namespace oxygen
```

7.7 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/Component.cc File Reference

```
#include "OxygenPCH.h"
#include "Component.h"
```

7.8 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/Component.h File Reference

Classes

- struct oxygen::Component

Namespaces

- namespace oxygen

7.9 Component.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     struct Entity;
00006     struct Component : ManagedObject
00007     {
00008         OXYGENOBJECT(Component, ManagedObject);
00009
00010         auto GetEntity() const -> std::shared_ptr<Entity>
00011         {
00012             return m_entity.lock();
00013         }
00014
00015         auto IsEnabled() const -> oxyBool
00016         {
00017             return m_enabled;
00018         }
00019         auto SetEnabled(oxyBool enabled) -> void
00020         {
00021             m_enabled = enabled;
00022         }
00023
00024       protected:
00025         virtual auto Update(oxyF32 deltaTimeSeconds) -> void{};
00026         virtual auto Render() const -> void{};
00027
00028       private:
00029         std::weak_ptr<Entity> m_entity{};
00030         oxyBool m_enabled{true};
00031
00032         friend struct Entity;
00033     };
00034 }; // namespace oxygen
```

7.10 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/EnvPushComponent/EnvPushComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "EnvPushComponent.h"
#include "World/World.h"
#include "Entity/Entity.h"
#include "Component/HullComponent/HullComponent.h"
```

Namespaces

- namespace oxygen

7.11 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/EnvPushComponent/EnvPushComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::EnvPushComponent

Namespaces

- namespace oxygen

7.12 EnvPushComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct EnvPushComponent final : Component
00008     {
00009         OXYGENOBJECT(EnvPushComponent, Component);
00010
00011         auto GetVelocity() const -> const oxyVec3&
00012         {
00013             return m_velocity;
00014         }
00015         auto GetRadius() const -> oxyF32
00016         {
00017             return m_radius;
00018         }
00019
00020         auto SetVelocity(const oxyVec3& velocity) -> void
00021         {
00022             m_velocity = velocity;
00023         }
00024         auto SetRadius(oxyF32 radius) -> void
00025         {
00026             m_radius = radius;
00027         }
00028
00029         protected:
00030         auto Update(oxyF32 deltaTimeSeconds) -> void override;
00031
00032       private:
00033         oxyVec3 m_velocity{};
00034         oxyF32 m_radius{};
00035         oxyBool m_isPushing{};
00036     };
00037 } // namespace oxygen
```

7.13 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/HealthComponent/HealthComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "HealthComponent.h"
#include "Net/NetSystem.h"
```

Namespaces

- namespace oxygen

7.14 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/HealthComponent/HealthComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::HealthComponent

Namespaces

- namespace oxygen

7.15 HealthComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct HealthComponent : Component
00008     {
00009         OXYGENOBJECT(HealthComponent, Component);
00010
00011         template <typename... TArgs>
00012         auto AddHealthStateChangedEvent(TArgs&&... args) -> void
00013         {
00014             m_healthStateChangedEvent.AddCallback(std::forward<TArgs>(args)...);
00015         }
00016
00017         template <typename... TArgs>
00018         auto AddHealedEvent(TArgs&&... args) -> void
00019         {
00020             m_healedEvent.AddCallback(std::forward<TArgs>(args)...);
00021         }
00022
00023         template <typename... TArgs>
00024         auto AddDamagedEvent(TArgs&&... args) -> void
00025         {
00026             m_damagedEvent.AddCallback(std::forward<TArgs>(args)...);
00027         }
00028
00029         auto Heal(oxyS32 amount) -> void;
00030         auto Damage(oxyS32 amount, DamageType type) -> void;
00031
00032         auto SetHealth(oxyU32 health) -> void
00033         {
00034             m_health = health;
00035         }
00036         auto SetMaxHealth(oxyU32 maxHealth) -> void
00037         {
00038             m_maxHealth = maxHealth;
00039         }
00040
00041         auto GetHealth() const -> oxyU32
00042         {
00043             return m_health;
00044         }
00045         auto GetMaxHealth() const -> oxyU32
00046         {
00047             return m_maxHealth;
00048         }
00049
00050         auto GetState() const -> HealthState
00051         {
00052             return m_state;
```

```
00053         }
00054
00055       private:
00056         oxyS32 m_health{};
00057         oxyS32 m_maxHealth{};
00058         HealthState m_state{};
00059
00060         // HealthComponent* this
00061         // Entity* owner entity
00062         // HealthState new state
00063         CallbackList<void, HealthComponent*, struct Entity*, HealthState>
00064             m_healthStateChangedEvent;
00065
00066         // HealthComponent* this
00067         // Entity* owner entity
00068         // oxyU32 amount of healing
00069         CallbackList<void, HealthComponent*, struct Entity*, oxyS32>
00070             m_healedEvent;
00071
00072         // HealthComponent* this
00073         // Entity* owner entity
00074         // oxyU32 amount of damage
00075         CallbackList<void, HealthComponent*, struct Entity*, oxyS32>
00076             m_damagedEvent;
00077
00078         auto HostSendHealthStateChange() -> void;
00079         auto ClientReceiveHealthStateChange(oxyS32 newhealth, oxyS32 newmax,
00080                                             HealthState newstate)
00081             -> void;
00082         friend struct GameManager; // ^^^^
00083
00084     };
00085 } // namespace oxygen
```

7.16 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/HullComponent/HullComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "HullComponent.h"
#include "Entity/Entity.h"
#include "World/World.h"
```

Namespaces

- namespace oxygen

7.17 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/HullComponent/HullComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::HullComponent

Namespaces

- namespace oxygen

7.18 HullComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct HullComponent final : Component
00008     {
00009         OXYGENOBJECT(HullComponent, Component);
00010
00011         template <typename... TArgs>
00012         auto AddCollideEvent(TArgs&&... args) -> void
00013         {
00014             m_onCollideEvent.AddCallback(std::forward<TArgs>(args)...);
00015         }
00016         template <typename... TArgs>
00017         auto AddBounceEvent(TArgs&&... args) -> void
00018         {
00019             m_onBounceEvent.AddCallback(std::forward<TArgs>(args)...);
00020         }
00021
00022         auto TraceLine(const oxyVec3& start, const oxyVec3& end,
00023                        oxyVec3& outPosition,
00024                        oxyVec3& outNormal) const -> oxyBool;
00025         auto CollidesWithHull(const oxyVec3& otherHullWorldPosition,
00026                               CollisionHull otherHull, oxyVec3& outPosition,
00027                               oxyVec3& outNormal) const -> oxyBool;
00028         auto IsWithinRadius(const oxyVec3& position,
00029                             oxyF32 radius) const -> oxyBool;
00030
00031         auto DoesIgnoreEntity(const struct Entity* entity) const -> oxyBool;
00032
00033         auto GetHull() const -> CollisionHull
00034         {
00035             return m_hull;
00036         }
00037         auto GetVelocity() const -> const oxyVec3&
00038         {
00039             return m_velocity;
00040         }
00041         auto GetGravityPerSecond() const -> oxyF32
00042         {
00043             return m_gravityPerSecond;
00044         }
00045         auto GetDrag() const -> oxyF32
00046         {
00047             return m_drag;
00048         }
00049         auto GetSolidToOtherHulls() const -> oxyBool
00050         {
00051             return m_solid;
00052         }
00053         auto GetBounceVelocityMultiplier() const -> oxyF32
00054         {
00055             return m_bounceVelocityMultiplier;
00056         }
00057         auto GetResponseType() const -> oxyBool
00058         {
00059             return m_response;
00060         }
00061
00062         auto SetHull(CollisionHull hull) -> void
00063         {
00064             m_hull = hull;
00065         }
00066         auto SetVelocity(const oxyVec3& velocity) -> void
00067         {
00068             m_velocity = velocity;
00069         }
00070         auto SetGravityPerSecond(oxyF32 gravityPerSecond) -> void
00071         {
00072             m_gravityPerSecond = gravityPerSecond;
00073         }
00074         auto SetDrag(oxyF32 drag) -> void
00075         {
00076             m_drag = drag;
00077         }
00078         auto SetSolidToOtherHulls(oxyBool solid) -> void
00079         {
00080             m_solid = solid;
00081         }
00082         auto
```

```
00083         SetBounceVelocityMultiplier(oxyF32 bounceVelocityMultiplier) -> void
00084         {
00085             m_bounceVelocityMultiplier = bounceVelocityMultiplier;
00086         }
00087         auto SetResponse(CollisionResponseType response) -> void
00088         {
00089             m_response = response;
00090         }
00091         auto
00092         AddToIgnoreList(const std::shared_ptr<struct Entity>& entity) -> void;
00093
00094       protected:
00095         auto Update(oxyF32 deltaTimeSeconds) -> void override;
00096         //auto Render() const -> void override;
00097
00098       private:
00099         auto UpdateSlide(oxyF32 deltaTimeSeconds) -> void;
00100         auto UpdateBounce(oxyF32 deltaTimeSeconds) -> void;
00101
00102         auto ClipToHullsAndUpdateWorldPosition(
00103             const oxyVec3& position, const oxyVec3& newPosition,
00104             const struct World* world, struct Entity* self) -> void;
00105
00106         // HullComponent* this hull
00107         // Entity* other hull entity
00108         // const oxyVec3& position of the collision
00109         // const oxyVec3& normal of the collision (magnitude equal to
00110         // penetration distance)
00111         CallbackList<void, HullComponent*, struct Entity*, const oxyVec3&,
00112                      const oxyVec3&>
00113             m_onCollideEvent;
00114         // HullComponent* this hull
00115         // Entity* other hull entity
00116         // const oxyVec3& position of the bounce
00117         CallbackList<void, HullComponent*, struct Entity*, const oxyVec3&>
00118             m_onBounceEvent;
00119
00120         CollisionHull m_hull;
00121         oxyVec3 m_velocity{};
00122         oxyF32 m_gravityPerSecond{};
00123         oxyF32 m_drag{};
00124         oxyBool m_solid{true};
00125         oxyF32 m_bounceVelocityMultiplier{1.f};
00126         CollisionResponseType m_response{
00127             CollisionResponseType::CollisionResponseType_Bounce};
00128         std::vector<std::weak_ptr<const struct Entity» m_ignoreEntities;
00129     };
00130 }; // namespace oxygen
```

7.19 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/Pawn/Pawn.cc File Reference

```
#include "OxygenPCH.h"
#include "Pawn.h"
#include "World/World.h"
#include "Entity/Entity.h"
#include "Component/HullComponent/HullComponent.h"
#include "Component/CameraComponent/CameraComponent.h"
#include "Component/AnimatedMeshComponent/AnimatedMeshComponent.h"
#include "Component/WeaponComponent/WeaponComponent.h"
#include "Component/HealthComponent/HealthComponent.h"
#include "Net/NetSystem.h"
#include "Input/InputManager.h"
#include "Gfx/GfxRenderer.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.20 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/Pawn/Pawn.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::Pawn

Namespaces

- namespace oxygen

Enumerations

- enum oxygen::PawnState { oxygen::PawnState_Ground , oxygen::PawnState_Void }
- enum oxygen::PawnStance { oxygen::PawnStance_Stand , oxygen::PawnStance_Crouch , oxygen::PawnStance_Prone }

7.21 Pawn.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     enum PawnState
00008     {
00009         PawnState_Ground,
00010         PawnState_Void,
00011     };
00012
00013     enum PawnStance
00014     {
00015         PawnStance_Stand,
00016         PawnStance_Crouch,
00017         PawnStance_Prone,
00018     };
00019
00020     struct Pawn final : Component
00021     {
00022         OXYGENOBJECT(Pawn, Component);
00023
00024         auto GetState() const -> PawnState
00025         {
00026             return m_state;
00027         }
00028         auto GetStance() const -> PawnStance
00029         {
00030             return m_stance;
00031         }
00032
00033         auto GetEquippedWeapon() const
00034             -> const std::shared_ptr<struct WeaponComponent>&
00035         {
00036             return m_equippedWeapon;
00037         }
00038
00039         auto GetEquippedRightHandedWeapon() const
00040             -> const std::shared_ptr<struct WeaponComponent>&
00041         {
```

```
00042             return m_rightHandEquippedWeapon;
00043         }
00044
00045       protected:
00046         auto Update(oxyF32 deltaTimeSeconds) -> void override;
00047         auto Render() const -> void override;
00048
00049       private:
00050         auto SetState(PawnState state) -> void
00051         {
00052             m_state = state;
00053         }
00054         auto SetStance(PawnStance stance) -> void
00055         {
00056             m_stance = stance;
00057         }
00058
00059         auto ParseInput() -> void;
00060
00061         auto GroundStateUpdate(oxyF32 deltaTimeSeconds,
00062                                struct Entity& ent) -> void;
00063         auto VoidStateUpdate(oxyF32 deltaTimeSeconds,
00064                              struct Entity& ent) -> void;
00065
00066         auto GetStanceVelocity() const -> oxyF32;
00067
00068         auto HealthStateChanged(struct HealthComponent* comp,
00069                                 struct Entity* ent, HealthState state) -> void;
00070
00071         oxyBool m_localControl{};
00072         oxyVec2 m_moveVector{};
00073         oxyVec2 m_lookVector{};
00074         oxyBool m_dropInputPressed{};
00075         oxyBool m_fireInputDown{};
00076         oxyBool m_fire2InputDown{};
00077         oxyBool m_reloadInputDown{};
00078         oxyF32 m_timeDead{};
00079
00080         PawnState m_state{PawnState_Ground};
00081         PawnStance m_stance{PawnStance_Stand};
00082
00083         std::shared_ptr<struct WeaponComponent> m_equippedWeapon;
00084         std::shared_ptr<struct WeaponComponent> m_rightHandEquippedWeapon;
00085         std::shared_ptr<struct WeaponComponent> m_lastDroppedWeapon;
00086         oxyF32 m_weaponDropHistoryClearTimer{};
00087         oxyVec3 m_lastGroundStateUpdatePosition{};
00088
00089         std::shared_ptr<struct AnimatedMeshComponent> m_thirdPersonMesh;
00090         std::shared_ptr<struct AnimatedMeshComponent> m_firstPersonMesh;
00091
00092         std::shared_ptr<struct HullComponent> m_hull;
00093         std::shared_ptr<struct CameraComponent> m_camera;
00094         std::shared_ptr<struct HealthComponent> m_health;
00095
00096         auto
00097         PickupWeapon(std::shared_ptr<struct WeaponComponent> weapon) -> void;
00098         auto DropWeaponNetWrap() -> void;
00099         auto DropWeaponImpl() -> void;
00100
00101         auto HostHullCollideEvent(HullComponent* hull, Entity* other,
00102                                   const oxyVec3& position,
00103                                   const oxyVec3& normal) -> void;
00104
00105         friend struct GameManager;
00106         friend struct WeaponComponent;
00107     };
00108 }; // namespace oxygen
```

7.22 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/PickupComponent/PickupComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "PickupComponent.h"
#include "Entity/Entity.h"
#include "Component/Pawn/Pawn.h"
#include "Component/HullComponent/HullComponent.h"
```

```
#include "Net/NetSystem.h"
```

Namespaces

- namespace oxygen

7.23 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/ProjectileComponent/ProjectileComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "ProjectileComponent.h"
#include "Entity/Entity.h"
#include "World/World.h"
#include "Component/HullComponent/HullComponent.h"
#include "Component/HealthComponent/HealthComponent.h"
#include "Net/NetSystem.h"
```

Namespaces

- namespace oxygen

7.24 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/ProjectileComponent/ProjectileComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::ProjectileComponent

Namespaces

- namespace oxygen

7.25 ProjectileComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct ProjectileComponent final : Component
00008     {
00009         OXYGENOBJECT(ProjectileComponent, Component);
00010
00011         auto SetBouncesLeft(oxyS32 bounces) -> void
00012         {
00013             m_bouncesRemaining = bounces;
00014         }
00015         auto SetDamage(oxyF32 damage) -> void
00016         {
00017             m_damage = damage;
00018         }
00019         auto SetDamageRadius(oxyF32 radius) -> void
00020         {
00021             m_damageRadius = radius;
00022         }
00023
00024       protected:
00025         auto Update(oxyF32 deltaTimeSeconds) -> void override;
00026
00027       private:
00028         auto SetHull(std::shared_ptr<struct HullComponent> hull) -> void;
00029         friend struct GameManager; // ^^^^
00030
00031         auto OnBounce(struct HullComponent* hull, struct Entity* other,
00032                       const oxyVec3& position) -> void;
00033
00034         auto Explode() -> void;
00035
00036         std::shared_ptr<struct HullComponent> m_hull;
00037         oxyS32 m_bouncesRemaining;
00038         oxyF32 m_damage;
00039         oxyF32 m_damageRadius;
00040         oxyVec3 m_spinEuler{10.f, 0.f, 0.f};
00041     };
00042 } // namespace oxygen
```

7.26 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/StaticMeshComponent/StaticMeshComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "StaticMeshComponent.h"
#include "Entity/Entity.h"
#include "Resources/ResourceManager.h"
#include "Resources/StaticMeshResource.h"
#include "Gfx/GfxRenderer.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.27 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/StaticMeshComponent/StaticMeshComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::StaticMeshComponent

Namespaces

- namespace oxygen

7.28 StaticMeshComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct StaticMeshResource;
00008
00009     struct StaticMeshComponent final : Component
00010     {
00011         OXYGENOBJECT(StaticMeshComponent, Component);
00012
00013         auto LoadByName(std::string_view name) -> oxyBool;
00014
00015         auto SetLocalOffset(const oxyVec3& offset) -> void
00016         {
00017             m_localOffset = offset;
00018         }
00019
00020       protected:
00021         auto Render() const -> void override;
00022
00023       private:
00024         oxyVec3 m_localOffset{};
00025         std::shared_ptr<const StaticMeshResource> m_resource;
00026         std::shared_ptr<const struct GfxTexture> m_texture;
00027     };
00028 }; // namespace oxygen
```

7.29 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Component/WeaponComponent/WeaponComponent.cc File Reference

```
#include "OxygenPCH.h"
#include "WeaponComponent.h"
#include "Entity/Entity.h"
#include "Component/Pawn/Pawn.h"
#include "Component/HullComponent/HullComponent.h"
#include "Component/CameraComponent/CameraComponent.h"
#include "GameManager/GameManager.h"
#include "Net/NetSystem.h"
```

Namespaces

- namespace oxygen

7.30 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Component/WeaponComponent/WeaponComponent.h File Reference

```
#include "Component/Component.h"
```

Classes

- struct oxygen::WeaponComponent

Namespaces

- namespace oxygen

7.31 WeaponComponent.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Component/Component.h"
00004
00005 namespace oxygen
00006 {
00007     struct WeaponComponent final : Component
00008     {
00009         OXYGENOBJECT(WeaponComponent, Component);
00010
00011         auto HasInfiniteReserve() const -> oxyBool
00012         {
00013             return m_infiniteReserve;
00014         }
00015         auto HasInfiniteClip() const -> oxyBool
00016         {
00017             return m_infiniteClip;
00018         }
00019         auto GetReserveAmmo() const -> oxyU32
00020         {
00021             return m_reserveAmmo;
00022         }
00023         auto GetClipAmmo() const -> oxyU32
00024         {
00025             return m_clipAmmo;
00026         }
00027         auto GetMaxClipAmmo() const -> oxyU32
00028         {
00029             return m_maxClipAmmo;
00030         }
00031         auto GetMaxReserveAmmo() const -> oxyU32
00032         {
00033             return m_maxReserveAmmo;
00034         }
00035         auto GetBulletsPerShot() const -> oxyU32
00036         {
00037             return m_bulletsPerShot;
00038         }
00039         auto GetRPM() const -> oxyU32
00040         {
00041             return m_rpm;
```

```
00042         }
00043         auto GetTimeToReload() const -> oxyF32
00044         {
00045             return m_timeToReload;
00046         }
00047         auto GetSpreadRadians() const -> oxyVec2
00048         {
00049             return m_spreadRadians;
00050         }
00051         auto GetFireType() const -> WeaponFireType
00052         {
00053             return m_fireType;
00054         }
00055         auto GetDestroyOnFire() const -> oxyBool
00056         {
00057             return m_destroyOnFire;
00058         }
00059         auto GetRightHanded() const -> oxyBool
00060         {
00061             return m_rightHanded;
00062         }
00063         auto GetCanDrop() const -> oxyBool
00064         {
00065             return m_canDrop;
00066         }
00067
00068         auto SetInfiniteReserve(oxyBool infinite) -> void
00069         {
00070             m_infiniteReserve = infinite;
00071         }
00072         auto SetInfiniteClip(oxyBool infinite) -> void
00073         {
00074             m_infiniteClip = infinite;
00075         }
00076         auto SetReserveAmmo(oxyU32 ammo) -> void
00077         {
00078             m_reserveAmmo = ammo;
00079         }
00080         auto SetClipAmmo(oxyU32 ammo) -> void
00081         {
00082             m_clipAmmo = ammo;
00083         }
00084         auto SetMaxClipAmmo(oxyU32 ammo) -> void
00085         {
00086             m_maxClipAmmo = ammo;
00087         }
00088         auto SetMaxReserveAmmo(oxyU32 ammo) -> void
00089         {
00090             m_maxReserveAmmo = ammo;
00091         }
00092         auto SetBulletsPerShot(oxyU32 bullets) -> void
00093         {
00094             m_bulletsPerShot = bullets;
00095         }
00096         auto SetRPM(oxyU32 rpm) -> void
00097         {
00098             m_rpm = rpm;
00099         }
00100         auto SetTimeToReload(oxyF32 time) -> void
00101         {
00102             m_timeToReload = time;
00103         }
00104         auto SetSpreadRadians(oxyVec2 spread) -> void
00105         {
00106             m_spreadRadians = spread;
00107         }
00108         auto SetFireType(WeaponFireType type) -> void
00109         {
00110             m_fireType = type;
00111         }
00112         auto SetDestroyOnFire(oxyBool destroy) -> void
00113         {
00114             m_destroyOnFire = destroy;
00115         }
00116         auto SetRightHanded(oxyBool right) -> void
00117         {
00118             m_rightHanded = right;
00119         }
00120         auto SetCanDrop(oxyBool canDrop) -> void
00121         {
00122             m_canDrop = canDrop;
00123         }
00124
00125       protected:
00126         auto Update(oxyF32 deltaTimeSeconds) -> void override;
00127
00128       private:
```

```
00129         oxyBool m_infiniteReserve{false};
00130         oxyBool m_infiniteClip{false};
00131         oxyU32 m_reserveAmmo{0};
00132         oxyU32 m_clipAmmo{0};
00133         oxyU32 m_maxClipAmmo{0};
00134         oxyU32 m_maxReserveAmmo{0};
00135         oxyU32 m_bulletsPerShot{1};
00136         oxyU32 m_rpm{60};
00137         oxyF32 m_timeToReload{};
00138         oxyVec2 m_spreadRadians{};
00139         WeaponFireType m_fireType{WeaponFireType_Count};
00140         oxyBool m_destroyOnFire{};
00141         oxyBool m_rightHanded{};
00142         oxyBool m_canDrop{};
00143 
00144         std::weak_ptr<struct Pawn> m_owner;
00145 
00146         oxyBool m_fireInputDown{};
00147         oxyBool m_fire2InputDown{};
00148         oxyBool m_reloadInputDown{};
00149 
00150         oxyF32 m_timeSinceLastShot{};
00151         oxyF32 m_reloadTimer{};
00152 
00153         oxyBool m_reloading{};
00154 
00155 
00156 
00157         oxyVec3 m_weaponFireDirectionEuler{};
00158 
00159         auto OnPickedUp(std::shared_ptr<struct Pawn> pawn) -> void;
00160         auto OnDropped() -> void;
00161         auto SetFireInputDown(oxyBool pressed) -> void;
00162         auto SetFire2InputDown(oxyBool pressed) -> void;
00163         auto SetReloadInputDown(oxyBool pressed) -> void;
00164         friend struct Pawn; // ^^^
00165 
00166         auto FireInDirectionFromPos(const oxyVec3& euler, const oxyVec3& pos) -> void;
00167 
00168         auto Fire() -> void;
00169         friend struct GameManager; // ^^^
00170         // ugh there's a lot of friends everywhere!!!!!!!!
00171 
00172         auto RandomSpreadAngles() const -> oxyVec2;
00173 
00174         auto BeginReload() -> void;
00175         auto ReloadEnded() -> void;
00176 
00177         auto ResetStateAndTimers() -> void;
00178     };
00179 }; // namespace oxygen
```

7.32 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Containers/SPSCQueue.h File Reference

Classes

- struct oxygen::SPSCQueue< T, N >

Namespaces

- namespace oxygen

7.33 SPSCQueue.h

Go to the documentation of this file.

```
00001 #pragma once
00002 
00003 namespace oxygen
```

```
00004 {
00005     template <typename T, int N> struct SPSCQueue
00006     {
00007         static_assert(N > 0, "Queue size must be greater than 0");
00008 
00009         SPSCQueue() : m_writePosition(0), m_readPosition(0)
00010         {
00011         }
00012 
00013         template <typename T> auto TryPush(T&& value) -> bool
00014         {
00015             const auto writeIndex =
00016                 m_writePosition.load(std::memory_order_relaxed);
00017             const auto nextWriteIndex = (writeIndex + 1) % N;
00018 
00019             if (nextWriteIndex ==
00020                 m_readPosition.load(std::memory_order_acquire))
00021                 return false;
00022 
00023             m_items[writeIndex] = std::forward<T>(value);
00024             m_writePosition.store(nextWriteIndex, std::memory_order_release);
00025             return true;
00026         }
00027 
00028         auto TryPop(T& valueOut) -> bool
00029         {
00030             const auto readIndex =
00031                 m_readPosition.load(std::memory_order_relaxed);
00032 
00033             if (readIndex == m_writePosition.load(std::memory_order_acquire))
00034                 return false;
00035 
00036             // IMPORTANT:
00037             // This pop CANNOT modify any data in the items array
00038             // god forbid there is a mutable member...
00039             valueOut = T{static_cast<const T&>(m_items[readIndex])};
00040             m_readPosition.store((readIndex + 1) % N,
00041                                  std::memory_order_release);
00042             return true;
00043         }
00044 
00045       private:
00046         alignas(128) std::array<T, N> m_items;
00047         std::atomic<int> m_writePosition;
00048         std::atomic<int> m_readPosition;
00049         oxyU8 m_padding[128];
00050     };
00051 }; // namespace oxygen
```

7.34 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Entity/Entity.cc File Reference

```
#include "OxygenPCH.h"
#include "Entity.h"
#include "Component/Component.h"
#include "World/World.h"
```

Namespaces

- namespace oxygen

7.35 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Entity/Entity.h File Reference

Classes

- struct oxygen::EntityHierarchy
- struct oxygen::Entity

Namespaces

- namespace oxygen

Enumerations

- enum oxygen::EntityFlags : oxyU32 {
  oxygen::EntityFlags_Disabled = 1 << 0 , oxygen::EntityFlags_Static = 1 << 1 , oxygen::EntityFlags_Dynamic = 1 << 2 , oxygen::EntityFlags_Renderable = 1 << 3 , oxygen::EntityFlags_HasHull = 1 << 4 , oxygen::EntityFlags_HasCamera = 1 << 5 , oxygen::EntityFlags_Replicated = 1 << 6 , oxygen::EntityFlags_IsLocalPlayer = 1 << 7 , oxygen::EntityFlags_EnableTransformReplication = 1 << 8 , oxygen::EntityFlags_EnableTransformInterpolation = 1 << 9 }

7.36 Entity.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     enum EntityFlags : oxyU32
00006     {
00007         EntityFlags_Disabled = 1 « 0,
00008         EntityFlags_Static = 1 « 1,
00009         EntityFlags_Dynamic = 1 « 2,
00010         EntityFlags_Renderable = 1 « 3,
00011         EntityFlags_HasHull = 1 « 4,
00012         EntityFlags_HasCamera = 1 « 5,
00013         EntityFlags_Replicated = 1 « 6,
00014         EntityFlags_IsLocalPlayer = 1 « 7,
00015         EntityFlags_EnableTransformReplication = 1 « 8,
00016         EntityFlags_EnableTransformInterpolation = 1 « 9,
00017     };
00018
00019     struct EntityHierarchy
00020     {
00021         std::shared_ptr<struct Entity> m_parent{};
00022         std::vector<std::shared_ptr<struct Entity» m_children;
00023         std::weak_ptr<struct Entity> m_self{};
00024     };
00025
00026     struct Component;
00027     struct Entity final : ManagedObject
00028     {
00029         OXYGENOBJECT(Entity, ManagedObject);
00030
00031         auto GetLocalPosition() const -> const oxyVec3&
00032         {
00033             return m_localPosition;
00034         }
00035         auto GetLocalRotation() const -> const oxyQuat&
00036         {
00037             return m_localRotation;
00038         }
00039         auto GetLocalScale() const -> const oxyVec3&
00040         {
00041             return m_localScale;
00042         }
00043         auto GetWorldPosition() const -> oxyVec3;
00044         auto GetWorldRotation() const -> oxyQuat;
00045         auto GetWorldScale() const -> oxyVec3;
00046         auto GetWorldTransformMatrix() const -> oxyMat4x4;
00047         auto GetFlag(EntityFlags flag) const -> oxyBool
00048         {
00049             return static_cast<std::underlying_type_t<EntityFlags>>(m_flags) &
00050                    static_cast<std::underlying_type_t<EntityFlags>>(flag);
00051         }
00052         auto GetWorld() const -> std::shared_ptr<struct World>
00053         {
00054             return m_world.lock();
00055         }
00056         auto GetParent() const -> std::shared_ptr<Entity>
```

```
00057         {
00058             return m_hierarchy.m_parent;
00059         }
00060         auto GetRenderOcclusionMin() const -> const oxyVec3&
00061         {
00062             return m_renderOcclusionMin;
00063         }
00064         auto GetRenderOcclusionMax() const -> const oxyVec3&
00065         {
00066             return m_renderOcclusionMax;
00067         }
00068         auto SetLocalPosition(const oxyVec3& position) -> void
00069         {
00070             m_localPosition = position;
00071         }
00072         auto SetLocalRotation(const oxyQuat& rotation) -> void
00073         {
00074             m_localRotation = rotation;
00075         }
00076         auto SetLocalScale(const oxyVec3& scale) -> void
00077         {
00078             m_localScale = scale;
00079         }
00080         auto SetWorldPosition(const oxyVec3& position) -> void;
00081         auto SetWorldRotation(const oxyQuat& rotation) -> void;
00082         auto SetWorldScale(const oxyVec3& scale) -> void;
00083         auto SetFlag(EntityFlags flag, oxyBool state) -> void
00084         {
00085             if (state)
00086                 m_flags = static_cast<EntityFlags>(
00087                     static_cast<std::underlying_type_t<EntityFlags>>(m_flags) |
00088                     static_cast<std::underlying_type_t<EntityFlags>>(flag));
00089             else
00090                 m_flags = static_cast<EntityFlags>(
00091                     static_cast<std::underlying_type_t<EntityFlags>>(m_flags) &
00092                     ~static_cast<std::underlying_type_t<EntityFlags»(flag));
00093         }
00094         auto SetRenderOcclusionMin(const oxyVec3& min) -> void
00095         {
00096             m_renderOcclusionMin = min;
00097         }
00098         auto SetRenderOcclusionMax(const oxyVec3& max) -> void
00099         {
00100             m_renderOcclusionMax = max;
00101         }
00102
00103         template <typename T>
00104         auto AddComponent(oxyObjectID id = 0) -> std::shared_ptr<T>
00105         {
00106             auto component =
00107                 ObjectManager::GetInstance().CreateManagedObject<T>(id);
00108             component->m_entity = GetHardRef<Entity>();
00109             m_components.push_back(component);
00110
00111             return component;
00112         }
00113         template <typename T> auto GetComponent() const -> std::shared_ptr<T>
00114         {
00115             for (auto& component : m_components)
00116             {
00117                 if (component->IsA<T>())
00118                 {
00119                     return std::static_pointer_cast<T>(component);
00120                 }
00121             }
00122             return nullptr;
00123         }
00124
00125         auto Destroy() -> void;
00126
00127         auto SetParent(std::shared_ptr<Entity> parent) -> void;
00128
00129         auto Update(oxyF32 deltaTimeSeconds) -> void;
00130         auto Render() const -> void;
00131
00132       private:
00133         oxyVec3 m_localPosition{};
00134         oxyQuat m_localRotation{0.f, 0.f, 0.f, 1.f};
00135         oxyVec3 m_localScale{1.f, 1.f, 1.f};
00136         EntityHierarchy m_hierarchy{};
00137         EntityFlags m_flags{};
00138         std::weak_ptr<struct World> m_world{};
00139         std::vector<std::shared_ptr<Component» m_components{};
00140         oxyVec3 m_renderOcclusionMin{};
00141         oxyVec3 m_renderOcclusionMax{};
00142
00143         friend struct World;
```

```
00144     };
00145 }; // namespace oxygen
```

7.37 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/GameManager/GameManager.cc File Reference

```
#include "OxygenPCH.h"
#include "GameManager.h"
#include "World/World.h"
#include "World/WorldLoader.h"
#include "Entity/Entity.h"
#include "Component/Pawn/Pawn.h"
#include "Component/AnimatedMeshComponent/AnimatedMeshComponent.h"
#include "Component/StaticMeshComponent/StaticMeshComponent.h"
#include "Component/WeaponComponent/WeaponComponent.h"
#include "Component/HullComponent/HullComponent.h"
#include "Component/CameraComponent/CameraComponent.h"
#include "Component/ProjectileComponent/ProjectileComponent.h"
#include "Component/HealthComponent/HealthComponent.h"
#include "Net/NetSystem.h"
#include "UI/UIManager.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.38 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/GameManager/GameManager.h File Reference

```
#include "Singleton/Singleton.h"
```

Classes

- struct oxygen::GameManager

Namespaces

- namespace oxygen

7.39 GameManager.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 namespace oxygen
00006 {
00007     struct World;
00008     struct GameManager : SingletonBase<GameManager>
00009     {
00010         GameManager();
00011
00012         auto Render() -> void;
00013         auto Update(float deltaTimeSeconds) -> void;
00014
00015         auto HostSummonEntity(EntitySpawnType type, const oxyVec3& pos,
00016                               const oxyQuat& rot) -> std::shared_ptr<struct Entity>;
00017
00018         auto HostGame(std::string worldName) -> void;
00019
00020       private:
00021         friend struct NetSystem;
00022         auto HandlePacket(struct NetConnection& conn, oxyU16 type,
00023                           std::span<const oxyU8> data) -> void;
00024
00025         auto HostHandlePacket(struct NetConnection& conn, oxyU16 type,
00026                               std::span<const oxyU8> data) -> void;
00027         auto ClientHandlePacket(struct NetConnection& conn, oxyU16 type,
00028                                 std::span<const oxyU8> data) -> void;
00029
00030         auto ClientDisconnectedFromHost() -> void;
00031
00032         auto HostNewPeerConnected(struct NetConnection& conn) -> void;
00033         auto HostPeerDisconnected(struct NetConnection& conn) -> void;
00034
00035         auto SpawnEntityInWorld(
00036             EntitySpawnType type, const oxyVec3& pos, const oxyQuat& rot,
00037             std::vector<oxyObjectID>& ids) -> std::shared_ptr<struct Entity>;
00038
00039         auto SendPeerEntityHistory(struct NetConnection& conn) -> void;
00040
00041         auto HostSendEntityTransforms() -> void;
00042         auto ClientSendEntityTransforms() -> void;
00043
00044         auto InterpolateEntityTransforms(float deltaTimeSeconds) -> void;
00045
00046
00047         struct PeerData
00048         {
00049             oxyBool m_loadedIn{};
00050             std::weak_ptr<struct Entity> m_localPlayer;
00051         };
00052         std::unordered_map<oxyU64, PeerData> m_peers;
00053
00054         struct InterpolateEntityTransformData
00055         {
00056             std::weak_ptr<struct Entity> m_entity;
00057             oxyVec3 m_latestPosition;
00058             oxyQuat m_latestRotation;
00059             oxyF32 m_timeSinceReceived{};
00060
00061
00062         };
00063         std::vector<InterpolateEntityTransformData>
00064             m_interpolateEntityTransforms;
00065
00066         std::string m_worldName;
00067         std::shared_ptr<World> m_world;
00068
00069         std::vector<std::tuple<EntitySpawnType, std::vector<oxyObjectID>, std::weak_ptr<struct
      Entity>>>
00070             m_entitySpawnHistory;
00071
00072         oxyF32 m_timeUntilNextGolfclubSpawn{3.0f};
00073     };
00074 }; // namespace oxygen
```

7.40 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.cc File Reference

```
#include "OxygenPCH.h"
#include "GfxRenderer.h"
#include "GfxSoftwareRasterize.inl"
#include "GameManager/GameManager.h"
#include "UI/UIManager.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.41 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxRenderer.h File Reference

```
#include "Singleton/Singleton.h"
```

Classes

- struct oxygen::GfxTexture
- struct oxygen::GfxVertex
- struct oxygen::GfxTri
- struct oxygen::GfxRenderer

Namespaces

- namespace oxygen
- namespace oxygen::GraphicsAbstraction

Enumerations

- enum oxygen::GfxRenderStrategy : oxyU8 { oxygen::GfxRenderStrategy_DirectToGPU = 0 , oxygen::GfxRenderStrategy_Softw , oxygen::GfxRenderStrategy_SoftwareDepthRasterizePreSortedOverlay , oxygen::GfxRenderStrategy_SoftwareDepthRasteriz }
- enum oxygen::GfxCullType { oxygen::GfxCullType_None = 0 , oxygen::GfxCullType_Backface = 1 , oxygen::GfxCullType_Frontface = 2 }

Functions

- auto oxygen::CullBackfaceTri (const GfxTri &tri) -> bool
- auto oxygen::CullFrontfaceTri (const GfxTri &tri) -> bool

7.42 GfxRenderer.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 namespace oxygen
00006 {
00007     namespace GraphicsAbstraction
00008     {
00009         struct Texture;
00010     }; // namespace GraphicsAbstraction
00011
00012     struct GfxTexture
00013     {
00014         oxyU32 m_width;
00015         oxyU32 m_height;
00016         std::string m_texturePath;
00017         std::shared_ptr<const GraphicsAbstraction::Texture> m_texture;
00018     };
00019
00020     struct GfxVertex
00021     {
00022         oxyVec4 m_position;
00023         oxyVec2 m_uv;
00024     };
00025     enum GfxRenderStrategy : oxyU8
00026     {
00027         // Submit straight to the GPU, 2D only, no depth writes or tests, no
00028         // clipping, drawn after all 3d geometry, in the order submitted
00029         // Usage: 2D sprites, UI elements
00030         GfxRenderStrategy_DirectToGPU = 0,
00031         // Submit to GPU, write depth buffer w/o test,
00032         // clips against all six planes
00033         // Usage: BSP sorted 3D geometry
00034         GfxRenderStrategy_SoftwareDepthRasterizePreSorted,
00035
00036         // Submit to GPU immediately after paired PreSorted
00037         // clips against all six planes
00038         // Usage: BSP sorted 3D geometry
00039         GfxRenderStrategy_SoftwareDepthRasterizePreSortedOverlay,
00040
00041         // Write depth buffer w/ test,
00042         // submit to gpu in spans at the end of the frame,
00043         // clips against near and far planes
00044         // Usage: dynamic 3D geometry
00045         GfxRenderStrategy_SoftwareDepthRasterize,
00046     };
00047     enum GfxCullType
00048     {
00049         GfxCullType_None = 0,
00050         GfxCullType_Backface = 1,
00051         GfxCullType_Frontface = 2,
00052     };
00053     struct GfxTri
00054     {
00055         GfxVertex m_vertices[3];
00056         oxyVec3 m_colour;
00057         const GfxTexture* m_texture{};
00058         GfxCullType m_cullType;
00059     };
00060
00061     inline auto CullBackfaceTri(const GfxTri& tri) -> bool
00062     {
00063         return (tri.m_vertices[1].m_position - tri.m_vertices[0].m_position)
00064                     .CrossProduct(tri.m_vertices[2].m_position -
00065                                   tri.m_vertices[0].m_position)
00066                     .z < 0;
00067     }
00068     inline auto CullFrontfaceTri(const GfxTri& tri) -> bool
00069     {
00070         return (tri.m_vertices[1].m_position - tri.m_vertices[0].m_position)
00071                     .CrossProduct(tri.m_vertices[2].m_position -
00072                                   tri.m_vertices[0].m_position)
00073                     .z > 0;
00074     }
00075
00076     struct GfxRenderer : SingletonBase<GfxRenderer>
00077     {
00078         GfxRenderer();
00079         auto SetViewProjectionMatrix(const oxyMat4x4& viewProjectionMatrix)
00080             -> void
00081         {
00082             m_viewProjectionMatrix = viewProjectionMatrix;
```

```
00083         }
00084         auto GetViewProjectionMatrix() const -> const oxyMat4x4&
00085         {
00086             return m_viewProjectionMatrix;
00087         }
00088
00089         auto GetWidth() const -> oxyS32
00090         {
00091             return m_width;
00092         }
00093         auto GetHeight() const -> oxyS32
00094         {
00095             return m_height;
00096         }
00097
00098         auto LoadTexture(std::string_view texturePath)
00099             -> std::shared_ptr<const GfxTexture>;
00100
00101         auto OverlayText(std::string_view text, oxyF32 blxndc, oxyF32 blyndc,
00102                          const oxyVec3& colour, oxyF32 spacing, oxyF32 size, oxyBool center) -> void;
00103         auto OverlayRect(const oxyVec3& col, const oxyVec2& minndc,
00104                          const oxyVec2& maxndc) -> void;
00105
00106         auto BeginFrame(oxyS32 w, oxyS32 h) -> void;
00107         auto EndFrame() -> void;
00108
00109         auto SubmitTriToQueue(const GfxTri& tri, GfxRenderStrategy mode, oxyF32 zmult = 1.0f)
00110             -> void;
00111       private:
00112         enum ClipCode
00113         {
00114             ClipCode_None = 0,
00115             ClipCode_Near = 1,
00116             ClipCode_Far = 2,
00117             ClipCode_Left = 4,
00118             ClipCode_Right = 8,
00119             ClipCode_Top = 16,
00120             ClipCode_Bottom = 32,
00121         };
00122         template <typename Fun>
00123         static auto ClipTri(const GfxTri& tri, ClipCode clipcode, Fun&& cb)
00124             -> void;
00125
00126         auto CullClipSpaceTri(const GfxTri& tri) -> bool;
00127
00128         auto ConvertTriToNDCAndCull(GfxTri& tri) -> bool;
00129
00130         auto DrawPreSortedTri(const GfxTri& tri) -> void;
00131
00132         auto DrawSpans(oxyU16 width, oxyU16 height) -> void;
00133         auto GetTriFromID(oxyS16 id) -> const GfxTri*;
00134         auto DrawSpan(const GfxTri& tri, oxyS32 y, oxyS32 x0, oxyS32 x1, oxyU16 fbwidth, oxyU16
    fbheight)
00135             -> void;
00136
00137         auto HandleResize(oxyS32 w, oxyS32 h) -> void;
00138         oxyS32 m_width;
00139         oxyS32 m_height;
00140
00141         oxyS32 m_softwareWidth;
00142         oxyS32 m_softwareHeight;
00143
00144         oxyU64 m_frameCounter{};
00145
00146         std::unique_ptr<oxyF32[]> m_zbuffer;
00147         std::unique_ptr<oxyS16[]> m_tribuffer;
00148
00149         oxyMat4x4 m_viewProjectionMatrix;
00150
00151         std::unordered_map<std::size_t, std::weak_ptr<const GfxTexture»
00152             m_textures;
00153
00154         std::shared_ptr<const GfxTexture> m_errorTexture;
00155         std::shared_ptr<const GfxTexture> m_whiteSolidTexture;
00156         std::shared_ptr<const GfxTexture> m_fontAtlasTexture;
00157         static inline constexpr auto k_fontAtlasBeginASCII = 32;
00158         static inline constexpr auto k_fontAtlasEndASCII = 127;
00159         static inline constexpr auto k_fontAtlasColumns = 16;
00160         static inline constexpr auto k_fontAtlasRows = 6;
00161
00162
00163         std::vector<GfxTri> m_triQueueSoftwareDepthRasterizePreSorted;
00164         std::vector<GfxTri> m_triQueueSoftwareDepthRasterizePreSortedOverlay;
00165         std::vector<GfxTri> m_triQueueSoftwareDepthRasterize;
00166         std::vector<GfxTri> m_triQueueDirectToGPU;
00167
00168         struct BBox
```

```
00169         {
00170             oxyS32 m_x0;
00171             oxyS32 m_y0;
00172             oxyS32 m_x1;
00173             oxyS32 m_y1;
00174             oxyF32 m_maxDepth;
00175 
00176             auto Overlaps(const BBox& other) const -> oxyBool;
00177             auto Expand(const BBox& other) -> oxyBool;
00178         };
00179 
00180         auto NDCTriToBBox(const GfxTri& tri) -> BBox;
00181     };
00182 }; // namespace oxygen
```

7.43 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Gfx/GfxSoftwareRasterize.inl File Reference

Classes

- struct oxygen::GfxSoftwareRasterizer::CountingIterator< T >

Namespaces

- namespace oxygen
- namespace oxygen::GfxSoftwareRasterizer

Functions

- auto oxygen::GfxSoftwareRasterizer::RasterTriDepthTest (const GfxTri &tri, oxyS16 triID, oxyU32 width, oxyU32 height, oxyF32 *zbuffer, oxyS16 *tribuffer, oxyU32 divminx, oxyU32 divminy, oxyU32 divmaxx, oxyU32 divmaxy) -> oxyBool
- auto oxygen::GfxSoftwareRasterizer::RasterTriNoDepthCompare (const GfxTri &tri, oxyU32 width, oxyU32 height, oxyF32 *zbuffer, oxyU32 divminx, oxyU32 divminy, oxyU32 divmaxx, oxyU32 divmaxy) -> void

7.44 GfxSoftwareRasterize.inl

Go to the documentation of this file.

```
00001 namespace oxygen
00002 {
00003     namespace GfxSoftwareRasterizer
00004     {
00005         template <typename T> struct CountingIterator
00006         {
00007             using iterator_category = std::random_access_iterator_tag;
00008             using value_type = T;
00009             using difference_type = T;
00010             using pointer = T*;
00011             using reference = T&;
00012             const T& operator*() const
00013             {
00014                 return m_value;
00015             }
00016             CountingIterator& operator++()
00017             {
00018                 ++m_value;
00019                 return *this;
00020             }
00021             CountingIterator operator++(int)
00022             {
00023                 auto temp = *this;
00024                 ++*this;
```

```
00025                 return temp;
00026             }
00027             CountingIterator& operator--()
00028             {
00029                 --m_value;
00030                 return *this;
00031             }
00032             CountingIterator operator--(int)
00033             {
00034                 auto temp = *this;
00035                 --*this;
00036                 return temp;
00037             }
00038             CountingIterator& operator+=(const T& lhs)
00039             {
00040                 m_value += lhs;
00041                 return *this;
00042             }
00043             CountingIterator& operator-=(const T& lhs)
00044             {
00045                 m_value -= lhs;
00046                 return *this;
00047             }
00048             CountingIterator operator+(const T& lhs) const
00049             {
00050                 auto temp = *this;
00051                 temp += lhs;
00052                 return temp;
00053             }
00054             CountingIterator operator-(const T& lhs) const
00055             {
00056                 auto temp = *this;
00057                 temp -= lhs;
00058                 return temp;
00059             }
00060             bool operator==(const CountingIterator& lhs) const
00061             {
00062                 return m_value == lhs.m_value;
00063             }
00064             bool operator!=(const CountingIterator& lhs) const
00065             {
00066                 return m_value != lhs.m_value;
00067             }
00068             const T operator+(const CountingIterator& lhs) const
00069             {
00070                 return m_value + lhs.m_value;
00071             }
00072             T operator-(const CountingIterator& lhs) const
00073             {
00074                 return m_value - lhs.m_value;
00075             }
00076             T m_value;
00077         };
00078
00079         inline auto RasterTriDepthTest(const GfxTri& tri, oxyS16 triID,
00080                                        oxyU32 width, oxyU32 height,
00081                                        oxyF32* zbuffer, oxyS16* tribuffer, oxyU32 divminx, oxyU32
      divminy, oxyU32 divmaxx, oxyU32 divmaxy)
00082             -> oxyBool
00083         {
00084             oxyBool rasteredany{};
00085             oxyVec2 screenSpaceVerts[3];
00086             for (auto i = 0; i < 3; ++i)
00087             {
00088                 const auto& vert = tri.m_vertices[i];
00089                 const auto x = (vert.m_position.x + 1.f) * 0.5f * width;
00090                 const auto y = (1.f - vert.m_position.y) * 0.5f * height;
00091                 screenSpaceVerts[i] = {std::ceilf(x), std::ceilf(y)};
00092             }
00093
00094             auto minx = std::max<oxyS16>(
00095                 std::min<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].x),
00096                                   static_cast<oxyS16>(screenSpaceVerts[1].x),
00097                                   static_cast<oxyS16>(screenSpaceVerts[2].x)}),
00098                 0);
00099             auto maxx = std::min<oxyS16>(
00100                 std::max<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].x),
00101                                   static_cast<oxyS16>(screenSpaceVerts[1].x),
00102                                   static_cast<oxyS16>(screenSpaceVerts[2].x)}),
00103                 width - 1);
00104             auto miny = std::max<oxyS16>(
00105                 std::min<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].y),
00106                                   static_cast<oxyS16>(screenSpaceVerts[1].y),
00107                                   static_cast<oxyS16>(screenSpaceVerts[2].y)}),
00108                 0);
00109             auto maxy = std::min<oxyS16>(
00110                 std::max<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].y),
```

```
00111                                   static_cast<oxyS16>(screenSpaceVerts[1].y),
00112                                   static_cast<oxyS16>(screenSpaceVerts[2].y)}),
00113                 height);
00114
00115             if ((maxx - minx) <= 0 || (maxy - miny) <= 0)
00116                 return false;
00117
00118             // Clamp by divide region
00119
00120             if (minx >= static_cast<oxyS16>(divmaxx))
00121                 return false;
00122             if (miny >= static_cast<oxyS16>(divmaxy))
00123                 return false;
00124             if (maxx <= static_cast<oxyS16>(divminx))
00125                 return false;
00126             if (maxy <= static_cast<oxyS16>(divminy))
00127                 return false;
00128             minx = std::max<oxyS16>(minx, divminx);
00129             miny = std::max<oxyS16>(miny, divminy);
00130             maxx = std::min<oxyS16>(maxx, divmaxx);
00131             maxy = std::min<oxyS16>(maxy, divmaxy);
00132
00133
00134             const auto x10 = screenSpaceVerts[1].x - screenSpaceVerts[0].x;
00135             const auto x21 = screenSpaceVerts[2].x - screenSpaceVerts[1].x;
00136             const auto x02 = screenSpaceVerts[0].x - screenSpaceVerts[2].x;
00137             const auto y10 = screenSpaceVerts[1].y - screenSpaceVerts[0].y;
00138             const auto y21 = screenSpaceVerts[2].y - screenSpaceVerts[1].y;
00139             const auto y02 = screenSpaceVerts[0].y - screenSpaceVerts[2].y;
00140
00141             const auto area = x21 * y02 - x02 * y21;
00142             const auto invArea = 1.f / area;
00143
00144             std::for_each(
00145                 std::execution::par_unseq, CountingIterator<int>{miny},
00146                 CountingIterator<int>{maxy}, [&](auto y) {
00147                     for (auto x = minx; x <= maxx; ++x)
00148                     {
00149                         const auto bw0cross =
00150                             x21 * (y - screenSpaceVerts[2].y) -
00151                             (x - screenSpaceVerts[2].x) * y21;
00152                         const auto bw1cross =
00153                             x02 * (y - screenSpaceVerts[0].y) -
00154                             (x - screenSpaceVerts[0].x) * y02;
00155                         const auto bw2cross =
00156                             x10 * (y - screenSpaceVerts[1].y) -
00157                             (x - screenSpaceVerts[1].x) * y10;
00158
00159                         const auto bw0crossbits =
00160                             std::bit_cast<oxyU32>(bw0cross);
00161                         const auto bw1crossbits =
00162                             std::bit_cast<oxyU32>(bw1cross);
00163                         const auto bw2crossbits =
00164                             std::bit_cast<oxyU32>(bw2cross);
00165
00166                         // If all sign bits are equal
00167                         if ((bw0crossbits & 0x80000000) ==
00168                                 (bw1crossbits & 0x80000000) &&
00169                             (bw1crossbits & 0x80000000) ==
00170                                 (bw2crossbits & 0x80000000))
00171                         {
00172                             const auto w0 = bw0cross * invArea;
00173                             const auto w1 = bw1cross * invArea;
00174                             const auto w2 = bw2cross * invArea;
00175
00176                             const auto z = w0 * tri.m_vertices[0].m_position.z +
00177                                            w1 * tri.m_vertices[1].m_position.z +
00178                                            w2 * tri.m_vertices[2].m_position.z;
00179                             const auto index = y * width + x;
00180                             if (zbuffer[index] > z)
00181                             {
00182                                 zbuffer[index] = z;
00183                                 tribuffer[index] = triID;
00184                                 rasteredany = true;
00185                             }
00186                         }
00187                     }
00188                 });
00189             return rasteredany;
00190         }
00191         inline auto RasterTriNoDepthCompare(const GfxTri& tri, oxyU32 width,
00192                                             oxyU32 height, oxyF32* zbuffer,
00193                                             oxyU32 divminx, oxyU32 divminy,
00194                                             oxyU32 divmaxx, oxyU32 divmaxy)
00195             -> void
00196         {
00197             oxyVec2 screenSpaceVerts[3];
```

```
00198             for (auto i = 0; i < 3; ++i)
00199             {
00200                 const auto& vert = tri.m_vertices[i];
00201                 const auto x = (vert.m_position.x + 1.f) * 0.5f * width;
00202                 const auto y = (1.f - vert.m_position.y) * 0.5f * height;
00203                 screenSpaceVerts[i] = {std::ceilf(x), std::ceilf(y)};
00204             }
00205 
00206             auto minx = std::max<oxyS16>(
00207                 std::min<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].x),
00208                                   static_cast<oxyS16>(screenSpaceVerts[1].x),
00209                                   static_cast<oxyS16>(screenSpaceVerts[2].x)}),
00210                 0);
00211             auto maxx = std::min<oxyS16>(
00212                 std::max<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].x),
00213                                   static_cast<oxyS16>(screenSpaceVerts[1].x),
00214                                   static_cast<oxyS16>(screenSpaceVerts[2].x)}),
00215                 width - 1);
00216             auto miny = std::max<oxyS16>(
00217                 std::min<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].y),
00218                                   static_cast<oxyS16>(screenSpaceVerts[1].y),
00219                                   static_cast<oxyS16>(screenSpaceVerts[2].y)}),
00220                 0);
00221             auto maxy = std::min<oxyS16>(
00222                 std::max<oxyS16>({static_cast<oxyS16>(screenSpaceVerts[0].y),
00223                                   static_cast<oxyS16>(screenSpaceVerts[1].y),
00224                                   static_cast<oxyS16>(screenSpaceVerts[2].y)}),
00225                 height);
00226 
00227             if ((maxx - minx) <= 0 || (maxy - miny) <= 0)
00228                 return;
00229 
00230             // Clamp by divide region
00231 
00232             if (minx >= static_cast<oxyS16>(divmaxx))
00233                 return;
00234             if (miny >= static_cast<oxyS16>(divmaxy))
00235                 return;
00236             if (maxx <= static_cast<oxyS16>(divminx))
00237                 return;
00238             if (maxy <= static_cast<oxyS16>(divminy))
00239                 return;
00240             minx = std::max<oxyS16>(minx, divminx);
00241             miny = std::max<oxyS16>(miny, divminy);
00242             maxx = std::min<oxyS16>(maxx, divmaxx);
00243             maxy = std::min<oxyS16>(maxy, divmaxy);
00244 
00245             const auto x10 = screenSpaceVerts[1].x - screenSpaceVerts[0].x;
00246             const auto x21 = screenSpaceVerts[2].x - screenSpaceVerts[1].x;
00247             const auto x02 = screenSpaceVerts[0].x - screenSpaceVerts[2].x;
00248             const auto y10 = screenSpaceVerts[1].y - screenSpaceVerts[0].y;
00249             const auto y21 = screenSpaceVerts[2].y - screenSpaceVerts[1].y;
00250             const auto y02 = screenSpaceVerts[0].y - screenSpaceVerts[2].y;
00251 
00252             const auto area = x21 * y02 - x02 * y21;
00253             const auto invArea = 1.f / area;
00254 
00255             std::for_each(
00256                 std::execution::par_unseq, CountingIterator<int>{miny},
00257                 CountingIterator<int>{maxy}, [&](auto y) {
00258                     for (auto x = minx; x <= maxx; ++x)
00259                     {
00260                         const auto bw0cross =
00261                             x21 * (y - screenSpaceVerts[2].y) -
00262                             (x - screenSpaceVerts[2].x) * y21;
00263                         const auto bw1cross =
00264                             x02 * (y - screenSpaceVerts[0].y) -
00265                             (x - screenSpaceVerts[0].x) * y02;
00266                         const auto bw2cross =
00267                             x10 * (y - screenSpaceVerts[1].y) -
00268                             (x - screenSpaceVerts[1].x) * y10;
00269 
00270                         const auto bw0crossbits =
00271                             std::bit_cast<oxyU32>(bw0cross);
00272                         const auto bw1crossbits =
00273                             std::bit_cast<oxyU32>(bw1cross);
00274                         const auto bw2crossbits =
00275                             std::bit_cast<oxyU32>(bw2cross);
00276 
00277                         // If all sign bits are equal
00278                         if ((bw0crossbits & 0x80000000) ==
00279                                 (bw1crossbits & 0x80000000) &&
00280                             (bw1crossbits & 0x80000000) ==
00281                                 (bw2crossbits & 0x80000000))
00282                         {
00283                             const auto w0 = bw0cross * invArea;
00284                             const auto w1 = bw1cross * invArea;
```

```
00285                             const auto w2 = bw2cross * invArea;
00286
00287                             const auto z = w0 * tri.m_vertices[0].m_position.z +
00288                                            w1 * tri.m_vertices[1].m_position.z +
00289                                            w2 * tri.m_vertices[2].m_position.z;
00290                             const auto index = y * width + x;
00291                             zbuffer[index] = z;
00292                         }
00293                     }
00294                 });
00295         }
00296     }; // namespace GfxSoftwareRasterizer
00297 };      // namespace oxygen
```

7.45 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Input/InputManager.cc File Reference

```
#include "OxygenPCH.h"
#include "InputManager.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.46 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Input/InputManager.h File Reference

```
#include "Singleton/Singleton.h"
```

Classes

- struct oxygen::InputManager

Namespaces

- namespace oxygen

7.47 InputManager.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 namespace oxygen
00006 {
00007     struct InputManager : SingletonBase<InputManager>
00008     {
00009         auto GetMousePosition() const -> const oxyVec2
00010         {
00011             return {m_mouseX, m_mouseY};
00012         }
00013         auto GetMouseDelta() const -> const oxyVec2
00014         {
00015             return {m_mouseDeltaX, m_mouseDeltaY};
00016         }
00017         auto IsKeyDown(KeyboardButton key) const -> oxyBool
00018         {
00019             return m_currentKeyStates[static_cast<size_t>(key)];
00020         }
00021         auto WasKeyDown(KeyboardButton key) const -> oxyBool
00022         {
00023             return m_previousKeyStates[static_cast<size_t>(key)];
00024         }
00025         auto IsMouseButtonDown(MouseButton button) const -> oxyBool
00026         {
00027             return m_currentMouseStates[static_cast<size_t>(button)];
00028         }
00029         auto WasMouseButtonDown(MouseButton button) const -> oxyBool
00030         {
00031             return m_previousMouseStates[static_cast<size_t>(button)];
00032         }
00033
00034         auto IsControllerConnected(oxyU8 controller) const -> oxyBool
00035         {
00036             OXYCHECK(controller < k_maxControllers);
00037             return m_controllerConnected[controller];
00038         }
00039
00040         auto IsControllerButtonDown(oxyU8 controller,
00041                                     ControllerButton button) const -> oxyBool
00042         {
00043             OXYCHECK(controller < k_maxControllers);
00044             return m_currentControllerStates[controller]
00045                                             [static_cast<size_t>(button)];
00046         }
00047         auto WasControllerButtonDown(oxyU8 controller,
00048                                      ControllerButton button) const -> oxyBool
00049         {
00050             OXYCHECK(controller < k_maxControllers);
00051             return m_previousControllerStates[controller]
00052                                              [static_cast<size_t>(button)];
00053         }
00054         auto GetControllerAxis(oxyU8 controller, ControllerAxis axis) const
00055             -> oxyF32
00056         {
00057             OXYCHECK(controller < k_maxControllers);
00058             return m_controllerAxisStates[controller]
00059                                          [static_cast<size_t>(axis)];
00060         }
00061         auto GetPreviousControllerAxis(oxyU8 controller,
00062                                        ControllerAxis axis) const -> oxyF32
00063         {
00064             OXYCHECK(controller < k_maxControllers);
00065             return m_previousControllerAxisStates[controller]
00066                                                  [static_cast<size_t>(axis)];
00067         }
00068
00069         auto SetCursorLock(oxyBool lock) -> void
00070         {
00071             m_lockCursor = lock;
00072         }
00073
00074         auto Update() -> void;
00075
00076       private:
00077         std::bitset<KeyboardButton_Count> m_currentKeyStates;
00078         std::bitset<KeyboardButton_Count> m_previousKeyStates;
00079         std::bitset<MouseButton_Count> m_currentMouseStates;
00080         std::bitset<MouseButton_Count> m_previousMouseStates;
00081         oxyF32 m_mouseX{};
00082         oxyF32 m_mouseY{};
```

```
00083         oxyF32 m_mouseDeltaX{};
00084         oxyF32 m_mouseDeltaY{};
00085
00086         bool m_lockCursor{false};
00087
00088         static constexpr auto k_maxControllers{4};
00089         std::array<std::bitset<ControllerButton_Count>, k_maxControllers>
00090             m_currentControllerStates;
00091         std::array<std::bitset<ControllerButton_Count>, k_maxControllers>
00092             m_previousControllerStates;
00093         std::array<std::array<oxyF32, ControllerAxis_Count>, k_maxControllers>
00094             m_controllerAxisStates;
00095         std::array<std::array<oxyF32, ControllerAxis_Count>, k_maxControllers>
00096             m_previousControllerAxisStates;
00097         std::array<oxyBool, k_maxControllers> m_controllerConnected;
00098     };
00099 }; // namespace oxygen
```

7.48 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/Defs.h File Reference

Classes

- struct oxyVec2
- struct oxyVec3
- struct oxyVec4
- struct oxyMat4x4
- struct oxyQuat

Namespaces

- namespace oxygen
- namespace oxygen::Math

Functions

- constexpr auto operator- (const oxyVec2 &a) -> oxyVec2
- constexpr auto operator+ (const oxyVec2 &a, const oxyVec2 &b) -> oxyVec2
- constexpr auto operator- (const oxyVec2 &a, const oxyVec2 &b) -> oxyVec2
- constexpr auto operator∗ (const oxyVec2 &a, const oxyVec2 &b) -> oxyVec2
- constexpr auto operator/ (const oxyVec2 &a, const oxyVec2 &b) -> oxyVec2
- constexpr auto operator∗ (const oxyVec2 &a, oxyF32 b) -> oxyVec2
- constexpr auto operator/ (const oxyVec2 &a, oxyF32 b) -> oxyVec2
- constexpr auto operator- (const oxyVec3 &a) -> oxyVec3
- constexpr auto operator+ (const oxyVec3 &a, const oxyVec3 &b) -> oxyVec3
- constexpr auto operator- (const oxyVec3 &a, const oxyVec3 &b) -> oxyVec3
- constexpr auto operator∗ (const oxyVec3 &a, const oxyVec3 &b) -> oxyVec3
- constexpr auto operator/ (const oxyVec3 &a, const oxyVec3 &b) -> oxyVec3
- constexpr auto operator∗ (const oxyVec3 &a, oxyF32 b) -> oxyVec3
- constexpr auto operator/ (const oxyVec3 &a, oxyF32 b) -> oxyVec3
- constexpr auto operator- (const oxyVec4 &a) -> oxyVec4
- constexpr auto operator+ (const oxyVec4 &a, const oxyVec4 &b) -> oxyVec4
- constexpr auto operator- (const oxyVec4 &a, const oxyVec4 &b) -> oxyVec4
- constexpr auto operator∗ (const oxyVec4 &a, const oxyVec4 &b) -> oxyVec4
- constexpr auto operator/ (const oxyVec4 &a, const oxyVec4 &b) -> oxyVec4
- constexpr auto operator∗ (const oxyVec4 &a, oxyF32 b) -> oxyVec4

- constexpr auto operator/ (const oxyVec4 &a, oxyF32 b) -> oxyVec4
- constexpr auto operator∗ (const oxyVec4 &a, const oxyMat4x4 &b) -> oxyVec4
- constexpr auto operator∗ (const oxyMat4x4 &a, const oxyMat4x4 &b) -> oxyMat4x4
- constexpr auto operator∗ (const oxyMat4x4 &a, oxyF32 b) -> oxyMat4x4
- constexpr auto operator∗ (const oxyMat4x4 &a, const oxyVec4 &b) -> oxyVec4
- constexpr auto operator∗ (const oxyQuat &a, const oxyQuat &b) -> oxyQuat
- constexpr auto operator∗ (const oxyQuat &a, const oxyVec3 &b) -> oxyVec3
- constexpr auto operator∗ (const oxyQuat &a, oxyF32 b) -> oxyQuat
- constexpr auto operator- (const oxyQuat &a) -> oxyQuat
- constexpr auto operator∗ (const oxyQuat &a, const oxyMat4x4 &b) -> oxyMat4x4
- constexpr auto oxygen::Math::Translate (const oxyMat4x4 &m, const oxyVec3 &v) -> oxyMat4x4
- constexpr auto oxygen::Math::Rotate (const oxyMat4x4 &m, const oxyQuat &q) -> oxyMat4x4
- auto oxygen::Math::Rotate (const oxyMat4x4 &m, oxyF32 angle, const oxyVec3 &axis) -> oxyMat4x4
- constexpr auto oxygen::Math::Scale (const oxyMat4x4 &m, const oxyVec3 &v) -> oxyMat4x4
- auto oxygen::Math::LookAt (const oxyVec3 &eye, const oxyVec3 &center, const oxyVec3 &up) -> oxyMat4x4
- auto oxygen::Math::Perspective (oxyF32 fovy, oxyF32 aspect, oxyF32 near, oxyF32 far) -> oxyMat4x4
- auto oxygen::Math::InverseMatrix (const oxyMat4x4 &m) -> oxyMat4x4
- auto oxygen::Math::RotationMatrixToEuler (const oxyMat4x4 &m) -> oxyVec3
- auto oxygen::Math::Slerp (const oxyQuat &a, const oxyQuat &b, oxyF32 t) -> oxyQuat
- auto oxygen::Math::AngleAxisToQuat (const oxyF32 angle, const oxyVec3 &axis) -> oxyQuat
- auto oxygen::Math::QuatToEulerAngles (const oxyQuat &q) -> oxyVec3
- auto oxygen::Math::EulerAnglesToQuat (const oxyVec3 &v) -> oxyQuat
- auto oxygen::Math::QuatLookAt (const oxyVec3 &position, const oxyVec3 &where) -> oxyQuat
- auto oxygen::Math::EulerForward (const oxyVec3 &euler) -> oxyVec3
- auto oxygen::Math::ToHalfFloat (oxyF32 x) -> oxyU16
- auto oxygen::Math::FromHalfFloat (oxyU16 x) -> oxyF32

7.48.1 Function Documentation

7.48.1.1 operator∗() [1/14]

```
auto operator* (
            const oxyMat4x4 & a,
            const oxyMat4x4 & b) -> oxyMat4x4  [inline], [constexpr]
```

7.48.1.2 operator∗() [2/14]

```
auto operator* (
            const oxyMat4x4 & a,
            const oxyVec4 & b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.3 operator∗() [3/14]

```
auto operator* (
            const oxyMat4x4 & a,
            oxyF32 b) -> oxyMat4x4  [inline], [constexpr]
```

7.48.1.4 operator∗() [4/14]

```
auto operator* (
            const oxyQuat & a,
            const oxyMat4x4 & b) -> oxyMat4x4  [inline], [constexpr]
```

7.48.1.5 operator∗() [5/14]

```
auto operator* (
            const oxyQuat & a,
            const oxyQuat & b) -> oxyQuat  [inline], [constexpr]
```

7.48.1.6 operator∗() [6/14]

```
auto operator* (
            const oxyQuat & a,
            const oxyVec3 & b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.7 operator∗() [7/14]

```
auto operator* (
            const oxyQuat & a,
            oxyF32 b) -> oxyQuat  [inline], [constexpr]
```

7.48.1.8 operator∗() [8/14]

```
auto operator* (
            const oxyVec2 & a,
            const oxyVec2 & b) -> oxyVec2  [inline], [constexpr]
```

7.48.1.9 operator∗() [9/14]

```
auto operator* (
            const oxyVec2 & a,
            oxyF32 b) -> oxyVec2  [inline], [constexpr]
```

7.48.1.10 operator∗() [10/14]

```
auto operator* (
            const oxyVec3 & a,
            const oxyVec3 & b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.11 operator∗() [11/14]

```
auto operator* (
            const oxyVec3 & a,
            oxyF32 b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.12 operator*() [12/14]

```
auto operator* (
            const oxyVec4 & a,
            const oxyMat4x4 & b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.13 operator*() [13/14]

```
auto operator* (
            const oxyVec4 & a,
            const oxyVec4 & b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.14 operator*() [14/14]

```
auto operator* (
            const oxyVec4 & a,
            oxyF32 b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.15 operator+() [1/3]

```
auto operator+ (
            const oxyVec2 & a,
            const oxyVec2 & b) -> oxyVec2  [inline], [constexpr]
```

7.48.1.16 operator+() [2/3]

```
auto operator+ (
            const oxyVec3 & a,
            const oxyVec3 & b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.17 operator+() [3/3]

```
auto operator+ (
            const oxyVec4 & a,
            const oxyVec4 & b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.18 operator-() [1/7]

```
auto operator- (
            const oxyQuat & a) -> oxyQuat  [inline], [constexpr]
```

7.48.1.19 operator-() [2/7]

```
auto operator- (
            const oxyVec2 & a) -> oxyVec2  [inline], [constexpr]
```

7.48.1.20 operator-() [3/7]

```
auto operator- (
            const oxyVec2 & a,
            const oxyVec2 & b) -> oxyVec2  [inline], [constexpr]
```

7.48.1.21 operator-() [4/7]

```
auto operator- (
            const oxyVec3 & a) -> oxyVec3  [inline], [constexpr]
```

7.48.1.22 operator-() [5/7]

```
auto operator- (
            const oxyVec3 & a,
            const oxyVec3 & b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.23 operator-() [6/7]

```
auto operator- (
            const oxyVec4 & a) -> oxyVec4  [inline], [constexpr]
```

7.48.1.24 operator-() [7/7]

```
auto operator- (
            const oxyVec4 & a,
            const oxyVec4 & b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.25 operator/() [1/6]

```
auto operator/ (
            const oxyVec2 & a,
            const oxyVec2 & b) -> oxyVec2  [inline], [constexpr]
```

7.48.1.26 operator/() [2/6]

```
auto operator/ (
            const oxyVec2 & a,
            oxyF32 b) -> oxyVec2  [inline], [constexpr]
```

7.48.1.27 operator/() [3/6]

```
auto operator/ (
            const oxyVec3 & a,
            const oxyVec3 & b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.28 operator/() [4/6]

```
auto operator/ (
            const oxyVec3 & a,
            oxyF32 b) -> oxyVec3  [inline], [constexpr]
```

7.48.1.29 operator/() [5/6]

```
auto operator/ (
            const oxyVec4 & a,
            const oxyVec4 & b) -> oxyVec4  [inline], [constexpr]
```

7.48.1.30 operator/() [6/6]

```
auto operator/ (
            const oxyVec4 & a,
            oxyF32 b) -> oxyVec4  [inline], [constexpr]
```

7.49 Defs.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 struct oxyVec2;
00004 struct oxyVec3;
00005 struct oxyVec4;
00006 struct oxyQuat;
00007 struct oxyMat4x4;
00008
00009 namespace oxygen
00010 {
00011     namespace Math
00012     {
00013         static inline constexpr auto k_pi = 3.14159265358979323846f;
00014         static inline constexpr auto k_twoPi = 2.0f * k_pi;
00015         static inline constexpr auto k_halfPi = 0.5f * k_pi;
00016
00017         static inline constexpr auto k_radToDeg = 180.0f / k_pi;
00018         static inline constexpr auto k_degToRad = k_pi / 180.0f;
00019     }; // namespace Math
00020 }; // namespace oxygen
00021
00022 inline constexpr auto operator-(const oxyVec2& a) -> oxyVec2;
00023 inline constexpr auto operator+(const oxyVec2& a, const oxyVec2& b) -> oxyVec2;
00024 inline constexpr auto operator-(const oxyVec2& a, const oxyVec2& b) -> oxyVec2;
00025 inline constexpr auto operator*(const oxyVec2& a, const oxyVec2& b) -> oxyVec2;
00026 inline constexpr auto operator/(const oxyVec2& a, const oxyVec2& b) -> oxyVec2;
00027 inline constexpr auto operator*(const oxyVec2& a, oxyF32 b) -> oxyVec2;
00028 inline constexpr auto operator/(const oxyVec2& a, oxyF32 b) -> oxyVec2;
00029
00030 inline constexpr auto operator-(const oxyVec3& a) -> oxyVec3;
00031 inline constexpr auto operator+(const oxyVec3& a, const oxyVec3& b) -> oxyVec3;
00032 inline constexpr auto operator-(const oxyVec3& a, const oxyVec3& b) -> oxyVec3;
00033 inline constexpr auto operator*(const oxyVec3& a, const oxyVec3& b) -> oxyVec3;
00034 inline constexpr auto operator/(const oxyVec3& a, const oxyVec3& b) -> oxyVec3;
00035 inline constexpr auto operator*(const oxyVec3& a, oxyF32 b) -> oxyVec3;
00036 inline constexpr auto operator/(const oxyVec3& a, oxyF32 b) -> oxyVec3;
00037
00038 inline constexpr auto operator-(const oxyVec4& a) -> oxyVec4;
00039 inline constexpr auto operator+(const oxyVec4& a, const oxyVec4& b) -> oxyVec4;
00040 inline constexpr auto operator-(const oxyVec4& a, const oxyVec4& b) -> oxyVec4;
00041 inline constexpr auto operator*(const oxyVec4& a, const oxyVec4& b) -> oxyVec4;
00042 inline constexpr auto operator/(const oxyVec4& a, const oxyVec4& b) -> oxyVec4;
00043 inline constexpr auto operator*(const oxyVec4& a, oxyF32 b) -> oxyVec4;
00044 inline constexpr auto operator/(const oxyVec4& a, oxyF32 b) -> oxyVec4;
00045 inline constexpr auto operator*(const oxyVec4& a,
00046                                 const oxyMat4x4& b) -> oxyVec4;
```

```
00047
00048 inline constexpr auto operator*(const oxyMat4x4& a,
00049                                 const oxyMat4x4& b) -> oxyMat4x4;
00050 inline constexpr auto operator*(const oxyMat4x4& a, oxyF32 b) -> oxyMat4x4;
00051 inline constexpr auto operator*(const oxyMat4x4& a,
00052                                 const oxyVec4& b) -> oxyVec4;
00053
00054 inline constexpr auto operator*(const oxyQuat& a, const oxyQuat& b) -> oxyQuat;
00055 inline constexpr auto operator*(const oxyQuat& a, const oxyVec3& b) -> oxyVec3;
00056 inline constexpr auto operator*(const oxyQuat& a, oxyF32 b) -> oxyQuat;
00057 inline constexpr auto operator-(const oxyQuat& a) -> oxyQuat;
00058 inline constexpr auto operator*(const oxyQuat& a,
00059                                 const oxyMat4x4& b) -> oxyMat4x4;
00060
00061 struct oxyVec2
00062 {
00063     constexpr oxyVec2() : x(0), y(0)
00064     {
00065     }
00066     constexpr oxyVec2(oxyF32 x, oxyF32 y) : x(x), y(y)
00067     {
00068     }
00069
00070     oxyF32 x, y;
00071
00072     constexpr auto MagnitudeSquared() const -> oxyF32
00073     {
00074         return x * x + y * y;
00075     }
00076     auto Magnitude() const -> oxyF32
00077     {
00078         return std::sqrtf(x * x + y * y);
00079     }
00080
00081     auto Normalized() const -> oxyVec2
00082     {
00083         oxyF32 mag = Magnitude();
00084         return {x / mag, y / mag};
00085     }
00086     auto Normalize() -> oxyVec2&
00087     {
00088         oxyF32 mag = Magnitude();
00089         x /= mag;
00090         y /= mag;
00091         return *this;
00092     }
00093
00094     constexpr auto DotProduct(const oxyVec2& other) const -> oxyF32
00095     {
00096         return x * other.x + y * other.y;
00097     }
00098
00099     constexpr auto CrossProduct(const oxyVec2& other) const -> oxyF32
00100     {
00101         return x * other.y - y * other.x;
00102     }
00103
00104     constexpr auto Conjugate() const -> oxyVec2
00105     {
00106         return {-x, -y};
00107     }
00108
00109     constexpr auto Inversed() const -> oxyVec2
00110     {
00111         return Conjugate() / (x * x + y * y);
00112     }
00113     constexpr auto Inverse() -> oxyVec2&
00114     {
00115         *this = Inversed();
00116         return *this;
00117     }
00118
00119     constexpr auto operator+=(const oxyVec2& other) -> oxyVec2&
00120     {
00121         *this = *this + other;
00122         return *this;
00123     }
00124     constexpr auto operator-=(const oxyVec2& other) -> oxyVec2&
00125     {
00126         *this = *this - other;
00127         return *this;
00128     }
00129     constexpr auto operator*=(const oxyVec2& other) -> oxyVec2&
00130     {
00131         *this = *this * other;
00132         return *this;
00133     }
```

```
00134     constexpr auto operator/=(const oxyVec2& other) -> oxyVec2&
00135     {
00136         *this = *this / other;
00137         return *this;
00138     }
00139     constexpr auto operator*=(oxyF32 other) -> oxyVec2&
00140     {
00141         *this = *this * other;
00142         return *this;
00143     }
00144     constexpr auto operator/=(oxyF32 other) -> oxyVec2&
00145     {
00146         *this = *this / other;
00147         return *this;
00148     }
00149 };
00150
00151 struct oxyVec3
00152 {
00153     constexpr oxyVec3() : x(0), y(0), z(0)
00154     {
00155     }
00156     constexpr oxyVec3(oxyF32 x, oxyF32 y, oxyF32 z) : x(x), y(y), z(z)
00157     {
00158     }
00159     constexpr oxyVec3(const oxyVec2& v, oxyF32 z) : x(v.x), y(v.y), z(z)
00160     {
00161     }
00162     oxyF32 x, y, z;
00163
00164     operator oxyVec2() const
00165     {
00166         return {x, y};
00167     }
00168
00169     constexpr auto MagnitudeSquared() const -> oxyF32
00170     {
00171         return x * x + y * y + z * z;
00172     }
00173     auto Magnitude() const -> oxyF32
00174     {
00175         return std::sqrtf(x * x + y * y + z * z);
00176     }
00177
00178     auto Normalized() const -> oxyVec3
00179     {
00180         const auto mag = Magnitude();
00181         return {x / mag, y / mag, z / mag};
00182     }
00183     auto Normalize() -> oxyVec3&
00184     {
00185         const auto mag = Magnitude();
00186         x /= mag;
00187         y /= mag;
00188         z /= mag;
00189         return *this;
00190     }
00191
00192     constexpr auto DotProduct(const oxyVec3& other) const -> oxyF32
00193     {
00194         return x * other.x + y * other.y + z * other.z;
00195     }
00196
00197     constexpr auto CrossProduct(const oxyVec3& other) const -> oxyVec3
00198     {
00199         return {y * other.z - z * other.y, z * other.x - x * other.z,
00200                 x * other.y - y * other.x};
00201     }
00202
00203     constexpr auto Conjugate() const -> oxyVec3
00204     {
00205         return {-x, -y, z};
00206     }
00207
00208     constexpr auto Inversed() const -> oxyVec3
00209     {
00210         return Conjugate() / (x * x + y * y + z * z);
00211     }
00212     constexpr auto Inverse() -> oxyVec3&
00213     {
00214         *this = Inversed();
00215         return *this;
00216     }
00217
00218     constexpr auto operator+=(const oxyVec3& other) -> oxyVec3&
00219     {
00220         *this = *this + other;
```

```
00221         return *this;
00222     }
00223     constexpr auto operator-=(const oxyVec3& other) -> oxyVec3&
00224     {
00225         *this = *this - other;
00226         return *this;
00227     }
00228     constexpr auto operator*=(const oxyVec3& other) -> oxyVec3&
00229     {
00230         *this = *this * other;
00231         return *this;
00232     }
00233     constexpr auto operator/=(const oxyVec3& other) -> oxyVec3&
00234     {
00235         *this = *this / other;
00236         return *this;
00237     }
00238     constexpr auto operator*=(oxyF32 other) -> oxyVec3&
00239     {
00240         *this = *this * other;
00241         return *this;
00242     }
00243     constexpr auto operator/=(oxyF32 other) -> oxyVec3&
00244     {
00245         *this = *this / other;
00246         return *this;
00247     }
00248 };
00249
00250 struct oxyVec4
00251 {
00252     constexpr oxyVec4() : x(0), y(0), z(0), w(0)
00253     {
00254     }
00255     constexpr oxyVec4(oxyF32 x, oxyF32 y, oxyF32 z, oxyF32 w)
00256         : x(x), y(y), z(z), w(w)
00257     {
00258     }
00259     constexpr oxyVec4(const oxyVec3& v, oxyF32 w) : x(v.x), y(v.y), z(v.z), w(w)
00260     {
00261     }
00262     constexpr oxyVec4(const oxyVec2& v, oxyF32 z, oxyF32 w)
00263         : x(v.x), y(v.y), z(z), w(w)
00264     {
00265     }
00266     oxyF32 x, y, z, w;
00267
00268     operator oxyVec3() const
00269     {
00270         return {x, y, z};
00271     }
00272     operator oxyVec2() const
00273     {
00274         return {x, y};
00275     }
00276
00277     constexpr auto MagnitudeSquared() const -> oxyF32
00278     {
00279         return x * x + y * y + z * z + w * w;
00280     }
00281     auto Magnitude() const -> oxyF32
00282     {
00283         return std::sqrtf(x * x + y * y + z * z + w * w);
00284     }
00285
00286     auto Normalized() const -> oxyVec4
00287     {
00288         const auto mag = Magnitude();
00289         return {x / mag, y / mag, z / mag, w / mag};
00290     }
00291     auto Normalize() -> oxyVec4&
00292     {
00293         const auto mag = Magnitude();
00294         x /= mag;
00295         y /= mag;
00296         z /= mag;
00297         w /= mag;
00298         return *this;
00299     }
00300
00301     constexpr auto DotProduct(const oxyVec4& other) const -> oxyF32
00302     {
00303         return x * other.x + y * other.y + z * other.z + w * other.w;
00304     }
00305
00306     constexpr auto Conjugate() const -> oxyVec4
00307     {
```

```
00308         return {-x, -y, -z, w};
00309     }
00310
00311     constexpr auto Inversed() const -> oxyVec4
00312     {
00313         return Conjugate() / (x * x + y * y + z * z + w * w);
00314     }
00315     constexpr auto Inverse() -> oxyVec4&
00316     {
00317         *this = Inversed();
00318         return *this;
00319     }
00320     constexpr auto CrossProduct(const oxyVec4& other) const -> oxyVec4
00321     {
00322         return {y * other.z - z * other.y, z * other.x - x * other.z,
00323                 x * other.y - y * other.x, 0};
00324     }
00325
00326     constexpr auto operator+=(const oxyVec4& other) -> oxyVec4&
00327     {
00328         *this = *this + other;
00329         return *this;
00330     }
00331     constexpr auto operator-=(const oxyVec4& other) -> oxyVec4&
00332     {
00333         *this = *this - other;
00334         return *this;
00335     }
00336     constexpr auto operator*=(const oxyVec4& other) -> oxyVec4&
00337     {
00338         *this = *this * other;
00339         return *this;
00340     }
00341     constexpr auto operator/=(const oxyVec4& other) -> oxyVec4&
00342     {
00343         *this = *this / other;
00344         return *this;
00345     }
00346     constexpr auto operator*=(oxyF32 other) -> oxyVec4&
00347     {
00348         *this = *this * other;
00349         return *this;
00350     }
00351     constexpr auto operator/=(oxyF32 other) -> oxyVec4&
00352     {
00353         *this = *this / other;
00354         return *this;
00355     }
00356 };
00357
00358 struct oxyMat4x4
00359 {
00360     oxyF32 m[4][4];
00361
00362     constexpr auto Determinant() const -> oxyF32
00363     {
00364         return m[0][0] * m[1][1] * m[2][2] * m[3][3] +
00365                m[0][0] * m[1][2] * m[2][3] * m[3][1] +
00366                m[0][0] * m[1][3] * m[2][1] * m[3][2] +
00367                m[0][1] * m[1][0] * m[2][3] * m[3][2] +
00368                m[0][1] * m[1][2] * m[2][0] * m[3][3] +
00369                m[0][1] * m[1][3] * m[2][2] * m[3][0] +
00370                m[0][2] * m[1][0] * m[2][1] * m[3][3] +
00371                m[0][2] * m[1][1] * m[2][3] * m[3][0] +
00372                m[0][2] * m[1][3] * m[2][0] * m[3][1] +
00373                m[0][3] * m[1][0] * m[2][2] * m[3][1] +
00374                m[0][3] * m[1][1] * m[2][0] * m[3][2] +
00375                m[0][3] * m[1][2] * m[2][1] * m[3][0] -
00376                m[0][0] * m[1][1] * m[2][3] * m[3][2] -
00377                m[0][0] * m[1][2] * m[2][1] * m[3][3] -
00378                m[0][0] * m[1][3] * m[2][2] * m[3][1] -
00379                m[0][1] * m[1][0] * m[2][2] * m[3][3] -
00380                m[0][1] * m[1][2] * m[2][3] * m[3][0] -
00381                m[0][1] * m[1][3] * m[2][0] * m[3][2] -
00382                m[0][2] * m[1][0] * m[2][3] * m[3][1] -
00383                m[0][2] * m[1][1] * m[2][0] * m[3][3] -
00384                m[0][2] * m[1][3] * m[2][1] * m[3][0] -
00385                m[0][3] * m[1][0] * m[2][1] * m[3][2] -
00386                m[0][3] * m[1][1] * m[2][2] * m[3][0] -
00387                m[0][3] * m[1][2] * m[2][0] * m[3][1];
00388     }
00389
00390     constexpr auto Transposed() const -> oxyMat4x4
00391     {
00392         oxyMat4x4 result;
00393         for (int i = 0; i < 4; i++)
00394         {
```

```
00395             for (int j = 0; j < 4; j++)
00396             {
00397                 result.m[i][j] = m[j][i];
00398             }
00399         }
00400         return result;
00401     }
00402     constexpr auto Transpose() -> oxyMat4x4&
00403     {
00404         *this = Transposed();
00405         return *this;
00406     }
00407
00408     constexpr auto operator[](int i) -> oxyF32*
00409     {
00410         return m[i];
00411     }
00412     constexpr auto operator[](int i) const -> const oxyF32*
00413     {
00414         return m[i];
00415     }
00416
00417     constexpr auto operator*=(const oxyMat4x4& other) -> oxyMat4x4&
00418     {
00419         *this = *this * other;
00420         return *this;
00421     }
00422
00423     static inline constexpr auto Identity() -> oxyMat4x4
00424     {
00425         oxyMat4x4 result{};
00426         result.m[0][0] = 1;
00427         result.m[1][1] = 1;
00428         result.m[2][2] = 1;
00429         result.m[3][3] = 1;
00430         return result;
00431     }
00432 };
00433
00434 struct oxyQuat
00435 {
00436     oxyF32 x{0.0f}, y{0.0f}, z{0.0f}, w{1.0f};
00437
00438     constexpr auto MagnitudeSquared() const -> oxyF32
00439     {
00440         return x * x + y * y + z * z + w * w;
00441     }
00442     auto Magnitude() const -> oxyF32
00443     {
00444         return std::sqrtf(x * x + y * y + z * z + w * w);
00445     }
00446
00447     auto Normalized() const -> oxyQuat
00448     {
00449         const auto mag = Magnitude();
00450         return {x / mag, y / mag, z / mag, w / mag};
00451     }
00452     auto Normalize() -> oxyQuat&
00453     {
00454         const auto mag = Magnitude();
00455         x /= mag;
00456         y /= mag;
00457         z /= mag;
00458         w /= mag;
00459         return *this;
00460     }
00461
00462     constexpr auto DotProduct(const oxyQuat& other) const -> oxyF32
00463     {
00464         return x * other.x + y * other.y + z * other.z + w * other.w;
00465     }
00466
00467     constexpr auto Conjugate() const -> oxyQuat
00468     {
00469         return {-x, -y, -z, w};
00470     }
00471
00472     auto Inversed() const -> oxyQuat
00473     {
00474         const auto conj = Conjugate();
00475         const auto magsq = MagnitudeSquared();
00476         return {conj.x / magsq, conj.y / magsq, conj.z / magsq, conj.w / magsq};
00477     }
00478     auto Inverse() -> oxyQuat&
00479     {
00480         *this = Inversed();
00481         return *this;
```

```
00482     }
00483
00484     constexpr auto operator*=(const oxyQuat& other) -> oxyQuat&
00485     {
00486         *this = *this * other;
00487         return *this;
00488     }
00489 };
00490
00491 inline constexpr auto operator-(const oxyVec2& a) -> oxyVec2
00492 {
00493     return {-a.x, -a.y};
00494 }
00495 inline constexpr auto operator+(const oxyVec2& a, const oxyVec2& b) -> oxyVec2
00496 {
00497     return {a.x + b.x, a.y + b.y};
00498 }
00499 inline constexpr auto operator-(const oxyVec2& a, const oxyVec2& b) -> oxyVec2
00500 {
00501     return {a.x - b.x, a.y - b.y};
00502 }
00503 inline constexpr auto operator*(const oxyVec2& a, const oxyVec2& b) -> oxyVec2
00504 {
00505     return {a.x * b.x, a.y * b.y};
00506 }
00507 inline constexpr auto operator/(const oxyVec2& a, const oxyVec2& b) -> oxyVec2
00508 {
00509     return {a.x / b.x, a.y / b.y};
00510 }
00511 inline constexpr auto operator*(const oxyVec2& a, oxyF32 b) -> oxyVec2
00512 {
00513     return {a.x * b, a.y * b};
00514 }
00515 inline constexpr auto operator/(const oxyVec2& a, oxyF32 b) -> oxyVec2
00516 {
00517     return {a.x / b, a.y / b};
00518 }
00519
00520 inline constexpr auto operator-(const oxyVec3& a) -> oxyVec3
00521 {
00522     return {-a.x, -a.y, -a.z};
00523 }
00524 inline constexpr auto operator+(const oxyVec3& a, const oxyVec3& b) -> oxyVec3
00525 {
00526     return {a.x + b.x, a.y + b.y, a.z + b.z};
00527 }
00528 inline constexpr auto operator-(const oxyVec3& a, const oxyVec3& b) -> oxyVec3
00529 {
00530     return {a.x - b.x, a.y - b.y, a.z - b.z};
00531 }
00532 inline constexpr auto operator*(const oxyVec3& a, const oxyVec3& b) -> oxyVec3
00533 {
00534     return {a.x * b.x, a.y * b.y, a.z * b.z};
00535 }
00536 inline constexpr auto operator/(const oxyVec3& a, const oxyVec3& b) -> oxyVec3
00537 {
00538     return {a.x / b.x, a.y / b.y, a.z / b.z};
00539 }
00540 inline constexpr auto operator*(const oxyVec3& a, oxyF32 b) -> oxyVec3
00541 {
00542     return {a.x * b, a.y * b, a.z * b};
00543 }
00544 inline constexpr auto operator/(const oxyVec3& a, oxyF32 b) -> oxyVec3
00545 {
00546     return {a.x / b, a.y / b, a.z / b};
00547 }
00548
00549 inline constexpr auto operator-(const oxyVec4& a) -> oxyVec4
00550 {
00551     return {-a.x, -a.y, -a.z, -a.w};
00552 }
00553 inline constexpr auto operator+(const oxyVec4& a, const oxyVec4& b) -> oxyVec4
00554 {
00555     return {a.x + b.x, a.y + b.y, a.z + b.z, a.w + b.w};
00556 }
00557 inline constexpr auto operator-(const oxyVec4& a, const oxyVec4& b) -> oxyVec4
00558 {
00559     return {a.x - b.x, a.y - b.y, a.z - b.z, a.w - b.w};
00560 }
00561 inline constexpr auto operator*(const oxyVec4& a, const oxyVec4& b) -> oxyVec4
00562 {
00563     return {a.x * b.x, a.y * b.y, a.z * b.z, a.w * b.w};
00564 }
00565 inline constexpr auto operator/(const oxyVec4& a, const oxyVec4& b) -> oxyVec4
00566 {
00567     return {a.x / b.x, a.y / b.y, a.z / b.z, a.w / b.w};
00568 }
```

```
00569 inline constexpr auto operator*(const oxyVec4& a, oxyF32 b) -> oxyVec4
00570 {
00571     return {a.x * b, a.y * b, a.z * b, a.w * b};
00572 }
00573 inline constexpr auto operator/(const oxyVec4& a, oxyF32 b) -> oxyVec4
00574 {
00575     return {a.x / b, a.y / b, a.z / b, a.w / b};
00576 }
00577 inline constexpr auto operator*(const oxyVec4& a, const oxyMat4x4& b) -> oxyVec4
00578 {
00579     return {
00580         a.x * b.m[0][0] + a.y * b.m[1][0] + a.z * b.m[2][0] + a.w * b.m[3][0],
00581         a.x * b.m[0][1] + a.y * b.m[1][1] + a.z * b.m[2][1] + a.w * b.m[3][1],
00582         a.x * b.m[0][2] + a.y * b.m[1][2] + a.z * b.m[2][2] + a.w * b.m[3][2],
00583         a.x * b.m[0][3] + a.y * b.m[1][3] + a.z * b.m[2][3] + a.w * b.m[3][3]};
00584 }
00585
00586 inline constexpr auto operator*(const oxyMat4x4& a,
00587                                 const oxyMat4x4& b) -> oxyMat4x4
00588 {
00589     oxyMat4x4 result;
00590     for (int i = 0; i < 4; i++)
00591     {
00592         for (int j = 0; j < 4; j++)
00593         {
00594             result.m[i][j] = a.m[i][0] * b.m[0][j] + a.m[i][1] * b.m[1][j] +
00595                              a.m[i][2] * b.m[2][j] + a.m[i][3] * b.m[3][j];
00596         }
00597     }
00598     return result;
00599 }
00600 inline constexpr auto operator*(const oxyMat4x4& a, oxyF32 b) -> oxyMat4x4
00601 {
00602     oxyMat4x4 result = a;
00603     for (int i = 0; i < 4; i++)
00604     {
00605         for (int j = 0; j < 4; j++)
00606         {
00607             result.m[i][j] *= b;
00608         }
00609     }
00610     return result;
00611 }
00612 inline constexpr auto operator*(const oxyMat4x4& a, const oxyVec4& b) -> oxyVec4
00613 {
00614     oxyVec4 result;
00615     for (int i = 0; i < 4; i++)
00616     {
00617         result.x += a.m[i][0] * b.x;
00618         result.y += a.m[i][1] * b.y;
00619         result.z += a.m[i][2] * b.z;
00620         result.w += a.m[i][3] * b.w;
00621     }
00622     return result;
00623 }
00624
00625 inline constexpr auto operator*(const oxyQuat& a, const oxyQuat& b) -> oxyQuat
00626 {
00627     return {a.w * b.x + a.x * b.w + a.y * b.z - a.z * b.y,
00628             a.w * b.y - a.x * b.z + a.y * b.w + a.z * b.x,
00629             a.w * b.z + a.x * b.y - a.y * b.x + a.z * b.w,
00630             a.w * b.w - a.x * b.x - a.y * b.y - a.z * b.z};
00631 }
00632 inline constexpr auto operator*(const oxyQuat& a, const oxyVec3& b) -> oxyVec3
00633 {
00634     oxyQuat v = {b.x, b.y, b.z, 0};
00635     oxyQuat result = a * v * a.Conjugate();
00636     return {result.x, result.y, result.z};
00637 }
00638 inline constexpr auto operator*(const oxyQuat& a, oxyF32 b) -> oxyQuat
00639 {
00640     return {a.x * b, a.y * b, a.z * b, a.w * b};
00641 }
00642 inline constexpr auto operator-(const oxyQuat& a) -> oxyQuat
00643 {
00644     return {-a.x, -a.y, -a.z, -a.w};
00645 }
00646 inline constexpr auto operator*(const oxyQuat& a,
00647                                 const oxyMat4x4& b) -> oxyMat4x4
00648 {
00649     oxyMat4x4 result;
00650     result.m[0][0] = 1 - 2 * a.y * a.y - 2 * a.z * a.z;
00651     result.m[0][1] = 2 * a.x * a.y - 2 * a.z * a.w;
00652     result.m[0][2] = 2 * a.x * a.z + 2 * a.y * a.w;
00653     result.m[0][3] = 0;
00654     result.m[1][0] = 2 * a.x * a.y + 2 * a.z * a.w;
00655     result.m[1][1] = 1 - 2 * a.x * a.x - 2 * a.z * a.z;
```

```
00656     result.m[1][2] = 2 * a.y * a.z - 2 * a.x * a.w;
00657     result.m[1][3] = 0;
00658     result.m[2][0] = 2 * a.x * a.z - 2 * a.y * a.w;
00659     result.m[2][1] = 2 * a.y * a.z + 2 * a.x * a.w;
00660     result.m[2][2] = 1 - 2 * a.x * a.x - 2 * a.y * a.y;
00661     result.m[2][3] = 0;
00662     result.m[3][0] = 0;
00663     result.m[3][1] = 0;
00664     result.m[3][2] = 0;
00665     result.m[3][3] = 1;
00666     return result * b;
00667 }
00668
00669 namespace oxygen
00670 {
00671     namespace Math
00672     {
00673         inline constexpr auto Translate(const oxyMat4x4& m,
00674                                         const oxyVec3& v) -> oxyMat4x4
00675         {
00676             oxyMat4x4 result = m;
00677             result.m[3][0] += v.x;
00678             result.m[3][1] += v.y;
00679             result.m[3][2] += v.z;
00680             return result;
00681         }
00682         inline constexpr auto Rotate(const oxyMat4x4& m,
00683                                      const oxyQuat& q) -> oxyMat4x4
00684         {
00685             oxyMat4x4 result;
00686             result.m[0][0] = 1 - 2 * q.y * q.y - 2 * q.z * q.z;
00687             result.m[0][1] = 2 * q.x * q.y - 2 * q.z * q.w;
00688             result.m[0][2] = 2 * q.x * q.z + 2 * q.y * q.w;
00689             result.m[0][3] = 0;
00690             result.m[1][0] = 2 * q.x * q.y + 2 * q.z * q.w;
00691             result.m[1][1] = 1 - 2 * q.x * q.x - 2 * q.z * q.z;
00692             result.m[1][2] = 2 * q.y * q.z - 2 * q.x * q.w;
00693             result.m[1][3] = 0;
00694             result.m[2][0] = 2 * q.x * q.z - 2 * q.y * q.w;
00695             result.m[2][1] = 2 * q.y * q.z + 2 * q.x * q.w;
00696             result.m[2][2] = 1 - 2 * q.x * q.x - 2 * q.y * q.y;
00697             result.m[2][3] = 0;
00698             result.m[3][0] = 0;
00699             result.m[3][1] = 0;
00700             result.m[3][2] = 0;
00701             result.m[3][3] = 1;
00702             return result * m;
00703         }
00704         inline auto Rotate(const oxyMat4x4& m, oxyF32 angle,
00705                            const oxyVec3& axis) -> oxyMat4x4
00706         {
00707             const auto halfAngle = angle / 2;
00708             const auto s = std::sinf(halfAngle);
00709             const auto q = oxyQuat{axis.x * s, axis.y * s, axis.z * s,
00710                                    std::cosf(halfAngle)};
00711             return Rotate(m, q);
00712         }
00713         inline constexpr auto Scale(const oxyMat4x4& m,
00714                                     const oxyVec3& v) -> oxyMat4x4
00715         {
00716             oxyMat4x4 result = m;
00717             result.m[0][0] *= v.x;
00718             result.m[1][1] *= v.y;
00719             result.m[2][2] *= v.z;
00720             return result;
00721         }
00722         inline auto LookAt(const oxyVec3& eye, const oxyVec3& center,
00723                            const oxyVec3& up) -> oxyMat4x4
00724         {
00725             const auto f = (center - eye).Normalized();
00726             const auto s = up.CrossProduct(f).Normalized();
00727             const auto u = f.CrossProduct(s);
00728             oxyMat4x4 result{};
00729             result[0][0] = s.x;
00730             result[1][0] = s.y;
00731             result[2][0] = s.z;
00732             result[0][1] = u.x;
00733             result[1][1] = u.y;
00734             result[2][1] = u.z;
00735             result[0][2] = f.x;
00736             result[1][2] = f.y;
00737             result[2][2] = f.z;
00738             result[3][0] = -s.DotProduct(eye);
00739             result[3][1] = -u.DotProduct(eye);
00740             result[3][2] = -f.DotProduct(eye);
00741             result[3][3] = 1;
00742             return result;
```

```
00743         }
00744         inline auto Perspective(oxyF32 fovy, oxyF32 aspect, oxyF32 near,
00745                                 oxyF32 far) -> oxyMat4x4
00746         {
00747             const auto tanHalfFovy = std::tanf(fovy / 2.f);
00748             oxyMat4x4 result{};
00749             result[0][0] = 1.f / (aspect * tanHalfFovy);
00750             result[1][1] = 1.f / tanHalfFovy;
00751             result[2][2] = far / (far - near);
00752             result[2][3] = 1.f;
00753             result[3][2] = -(far * near) / (far - near);
00754             return result;
00755         }
00756         inline auto InverseMatrix(const oxyMat4x4& m) -> oxyMat4x4
00757         {
00758             oxyMat4x4 result;
00759             const auto det = m.Determinant();
00760             if (det == 0)
00761                 return result;
00762             const auto invDet = 1 / det;
00763             result[0][0] =
00764                 invDet *
00765                 (m[1][1] * m[2][2] * m[3][3] + m[1][2] * m[2][3] * m[3][1] +
00766                  m[1][3] * m[2][1] * m[3][2] - m[1][1] * m[2][3] * m[3][2] -
00767                  m[1][2] * m[2][1] * m[3][3] - m[1][3] * m[2][2] * m[3][1]);
00768             result[0][1] =
00769                 invDet *
00770                 (m[0][1] * m[2][3] * m[3][2] + m[0][2] * m[2][1] * m[3][3] +
00771                  m[0][3] * m[2][2] * m[3][1] - m[0][1] * m[2][2] * m[3][3] -
00772                  m[0][2] * m[2][3] * m[3][1] - m[0][3] * m[2][1] * m[3][2]);
00773             result[0][2] =
00774                 invDet *
00775                 (m[0][1] * m[1][2] * m[3][3] + m[0][2] * m[1][3] * m[3][1] +
00776                  m[0][3] * m[1][1] * m[3][2] - m[0][1] * m[1][3] * m[3][2] -
00777                  m[0][2] * m[1][1] * m[3][3] - m[0][3] * m[1][2] * m[3][1]);
00778             result[0][3] =
00779                 invDet *
00780                 (m[0][1] * m[1][3] * m[2][2] + m[0][2] * m[1][1] * m[2][3] +
00781                  m[0][3] * m[1][2] * m[2][1] - m[0][1] * m[1][2] * m[2][3] -
00782                  m[0][2] * m[1][3] * m[2][1] - m[0][3] * m[1][1] * m[2][2]);
00783
00784             result[1][0] =
00785                 invDet *
00786                 (m[1][0] * m[2][3] * m[3][2] + m[1][2] * m[2][0] * m[3][3] +
00787                  m[1][3] * m[2][2] * m[3][0] - m[1][0] * m[2][2] * m[3][3] -
00788                  m[1][2] * m[2][3] * m[3][0] - m[1][3] * m[2][0] * m[3][2]);
00789             result[1][1] =
00790                 invDet *
00791                 (m[0][0] * m[2][2] * m[3][3] + m[0][2] * m[2][3] * m[3][0] +
00792                  m[0][3] * m[2][0] * m[3][2] - m[0][0] * m[2][3] * m[3][2] -
00793                  m[0][2] * m[2][0] * m[3][3] - m[0][3] * m[2][2] * m[3][0]);
00794             result[1][2] =
00795                 invDet *
00796                 (m[0][0] * m[1][3] * m[3][2] + m[0][2] * m[1][0] * m[3][3] +
00797                  m[0][3] * m[1][2] * m[3][0] - m[0][0] * m[1][2] * m[3][3] -
00798                  m[0][2] * m[1][3] * m[3][0] - m[0][3] * m[1][0] * m[3][2]);
00799             result[1][3] =
00800                 invDet *
00801                 (m[0][0] * m[1][2] * m[2][3] + m[0][1] * m[1][3] * m[2][0] +
00802                  m[0][3] * m[1][0] * m[2][2] - m[0][0] * m[1][3] * m[2][2] -
00803                  m[0][1] * m[1][0] * m[2][3] - m[0][3] * m[1][2] * m[2][0]);
00804
00805             result[2][0] =
00806                 invDet *
00807                 (m[1][0] * m[2][3] * m[3][1] + m[1][1] * m[2][0] * m[3][3] +
00808                  m[1][3] * m[2][1] * m[3][0] - m[1][0] * m[2][1] * m[3][3] -
00809                  m[1][1] * m[2][3] * m[3][0] - m[1][3] * m[2][0] * m[3][1]);
00810             result[2][1] =
00811                 invDet *
00812                 (m[0][0] * m[2][1] * m[3][3] + m[0][1] * m[2][3] * m[3][0] +
00813                  m[0][3] * m[2][0] * m[3][1] - m[0][0] * m[2][3] * m[3][1] -
00814                  m[0][1] * m[2][0] * m[3][3] - m[0][3] * m[2][1] * m[3][0]);
00815             result[2][2] =
00816                 invDet *
00817                 (m[0][0] * m[1][1] * m[3][3] + m[0][1] * m[1][3] * m[3][0] +
00818                  m[0][3] * m[1][0] * m[3][1] - m[0][0] * m[1][3] * m[3][1] -
00819                  m[0][1] * m[1][0] * m[3][3] - m[0][3] * m[1][1] * m[3][0]);
00820             result[2][3] =
00821                 invDet *
00822                 (m[0][0] * m[1][1] * m[2][3] + m[0][1] * m[1][3] * m[2][0] +
00823                  m[0][2] * m[1][0] * m[2][1] - m[0][0] * m[1][3] * m[2][1] -
00824                  m[0][1] * m[1][0] * m[2][3] - m[0][2] * m[1][1] * m[2][0]);
00825
00826             result[3][0] =
00827                 invDet *
00828                 (m[1][0] * m[2][1] * m[3][2] + m[1][1] * m[2][2] * m[3][0] +
00829                  m[1][2] * m[2][0] * m[3][1] - m[1][0] * m[2][2] * m[3][1] -
```

```
00830                m[1][1] * m[2][0] * m[3][2] - m[1][2] * m[2][1] * m[3][0]);
00831            result[3][1] =
00832                invDet *
00833                (m[0][0] * m[2][2] * m[3][1] + m[0][1] * m[2][0] * m[3][2] +
00834                 m[0][2] * m[2][1] * m[3][0] - m[0][0] * m[2][1] * m[3][2] -
00835                 m[0][1] * m[2][2] * m[3][0] - m[0][2] * m[2][0] * m[3][1]);
00836            result[3][2] =
00837                invDet *
00838                (m[0][0] * m[1][1] * m[3][0] + m[0][1] * m[1][0] * m[3][2] +
00839                 m[0][2] * m[1][2] * m[3][1] - m[0][0] * m[1][2] * m[3][1] -
00840                 m[0][1] * m[1][2] * m[3][0] - m[0][2] * m[1][0] * m[3][1]);
00841            result[3][3] =
00842                invDet *
00843                (m[0][0] * m[1][1] * m[2][0] + m[0][1] * m[1][0] * m[2][1] +
00844                 m[0][2] * m[1][1] * m[2][2] - m[0][0] * m[1][1] * m[2][1] -
00845                 m[0][1] * m[1][2] * m[2][0] - m[0][2] * m[1][0] * m[2][1]);
00846            return result;
00847        }
00848        inline auto RotationMatrixToEuler(const oxyMat4x4& m) -> oxyVec3
00849        {
00850            oxyVec3 result;
00851            result.y = std::asin(-m[2][0]);
00852            if (std::cos(result.y) != 0)
00853            {
00854                result.x = std::atan2(m[2][1], m[2][2]);
00855                result.z = std::atan2(m[1][0], m[0][0]);
00856            }
00857            else
00858            {
00859                result.x = 0;
00860                result.z = std::atan2(-m[0][1], m[1][1]);
00861            }
00862            return result;
00863        }
00864        inline auto Slerp(const oxyQuat& a, const oxyQuat& b,
00865                          oxyF32 t) -> oxyQuat
00866        {
00867            const auto dot = a.DotProduct(b);
00868            const auto angle = std::acosf(dot);
00869            const auto denom = std::sinf(angle);
00870            if (denom == 0)
00871                return a;
00872            const auto s0 = std::sinf((1 - t) * angle) / denom;
00873            const auto s1 = std::sinf(t * angle) / denom;
00874            const auto sa = a * s0;
00875            const auto sb = b * s1;
00876            return oxyQuat{sa.x + sb.x, sa.y + sb.y, sa.z + sb.z, sa.w + sb.w};
00877        }
00878        inline auto AngleAxisToQuat(const oxyF32 angle,
00879                                    const oxyVec3& axis) -> oxyQuat
00880        {
00881            const auto halfAngle = angle / 2;
00882            const auto s = std::sinf(halfAngle);
00883            return {axis.x * s, axis.y * s, axis.z * s, std::cos(halfAngle)};
00884        }
00885        inline auto QuatToEulerAngles(const oxyQuat& q) -> oxyVec3
00886        {
00887            oxyVec3 result;
00888            result.x = std::atan2(2 * (q.w * q.x + q.y * q.z),
00889                                  1 - 2 * (q.x * q.x + q.y * q.y));
00890            result.y = std::asin(2 * (q.w * q.y - q.z * q.x));
00891            result.z = std::atan2(2 * (q.w * q.z + q.x * q.y),
00892                                  1 - 2 * (q.y * q.y + q.z * q.z));
00893            return result;
00894        }
00895        inline auto EulerAnglesToQuat(const oxyVec3& v) -> oxyQuat
00896        {
00897            const auto pitch = AngleAxisToQuat(v.x, {1, 0, 0});
00898            const auto yaw = AngleAxisToQuat(v.y, {0, 1, 0});
00899            const auto roll = AngleAxisToQuat(v.z, {0, 0, 1});
00900            return pitch * yaw * roll;
00901        }
00902        inline auto QuatLookAt(const oxyVec3& position,
00903                               const oxyVec3& where) -> oxyQuat
00904        {
00905            const auto direction = (where - position).Normalized();
00906            const auto forward = oxyVec3{0, 0, 1};
00907            const auto axis = forward.CrossProduct(direction);
00908            const auto angle = std::acosf(forward.DotProduct(direction));
00909            return AngleAxisToQuat(angle, axis);
00910        }
00911        inline auto EulerForward(const oxyVec3& euler) -> oxyVec3
00912        {
00913            return oxyVec3{std::cosf(euler.z) * std::cosf(euler.x),
00914                           std::sinf(euler.z) * std::cosf(euler.x),
00915                           std::sinf(euler.x)};
00916        }
```

```
00917         inline auto ToHalfFloat(oxyF32 x) -> oxyU16
00918         {
00919             uint32_t i = std::bit_cast<uint32_t>(x);
00920             uint16_t bits = (i » 16) & 0x8000;
00921             uint16_t m = (i » 12) & 0x7ff;
00922             uint16_t e = ((i » 23) & 0xff) - 127;
00923             if (e > 30)
00924             {
00925                 bits |= 0x7c00;
00926             }
00927             else if (e > 15)
00928             {
00929                 m |= 0x800;
00930                 bits |= (e « 10) | (m » 1);
00931             }
00932             else if (e > -15)
00933             {
00934                 bits |= ((e + 15) « 10) | (m » 1);
00935             }
00936             else if (e > -25)
00937             {
00938                 m |= 0x800;
00939                 bits |= ((e + 15) « 10) | (m » 1);
00940             }
00941             else
00942             {
00943                 bits |= 0;
00944             }
00945             return bits;
00946         }
00947         inline auto FromHalfFloat(oxyU16 x) -> oxyF32
00948         {
00949             uint32_t mantissa = x & 0x3ff;
00950             uint32_t exponent = x & 0x7c00;
00951             if (exponent == 0x7c00)
00952             {
00953                 exponent = 0x8f « 23;
00954             }
00955             else if (exponent == 0)
00956             {
00957                 if (mantissa != 0)
00958                 {
00959                     exponent = 0x71 « 23;
00960                     while ((mantissa & 0x400) == 0)
00961                     {
00962                         mantissa «= 1;
00963                         exponent -= 0x800000;
00964                     }
00965                     mantissa &= 0x3ff;
00966                 }
00967             }
00968             else
00969             {
00970                 exponent = (exponent » 10) + 0x70;
00971             }
00972             uint32_t result = (x & 0x8000) « 16;
00973             result |= (exponent « 23);
00974             result |= (mantissa « 13);
00975             return std::bit_cast<float>(result);
00976         }
00977
00978     }; // namespace Math
00979 }; // namespace oxygen
```

7.50 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/↩ Hash.h File Reference

Namespaces

- namespace oxygen

Functions

- constexpr auto oxygen::CRC64Eval (const oxyU8 *data, oxySize size) -> oxyU64

Variables

- constexpr oxyU64 oxygen::g_CRC64Table [ ]

7.51 Hash.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     inline constexpr oxyU64 g_CRC64Table[]{
00006         0x0000000000000000ULL, 0x42f0e1eba9ea3693ULL, 0x85e1c3d753d46d26ULL,
00007         0xc711223cfa3e5bb5ULL, 0x493366450e42ecdfULL, 0x0bc387aea7a8da4cULL,
00008         0xccd2a5925d9681f9ULL, 0x8e224479f47cb76aULL, 0x9266cc8a1c85d9beULL,
00009         0xd0962d61b56fef2dULL, 0x17870f5d4f51b498ULL, 0x5577eeb6e6bb820bULL,
00010         0xdb55aacf12c73561ULL, 0x99a54b24bb2d03f2ULL, 0x5eb4691841135847ULL,
00011         0x1c4488f3e8f96ed4ULL, 0x663d78ff90e185efULL, 0x24cd9914390bb37cULL,
00012         0xe3dcbb28c335e8c9ULL, 0xa12c5ac36adfde5aULL, 0x2f0e1eba9ea36930ULL,
00013         0x6dfeff5137495fa3ULL, 0xaaefdd6dcd770416ULL, 0xe81f3c86649d3285ULL,
00014         0xf45bb4758c645c51ULL, 0xb6ab559e258e6ac2ULL, 0x71ba77a2dfb03177ULL,
00015         0x334a9649765a07e4ULL, 0xbd68d2308226b08eULL, 0xff9833db2bcc861dULL,
00016         0x388911e7d1f2dda8ULL, 0x7a79f00c7818eb3bULL, 0xcc7af1ff21c30bdeULL,
00017         0x8e8a101488293d4dULL, 0x499b3228721766f8ULL, 0x0b6bd3c3dbfd506bULL,
00018         0x854997ba2f81e701ULL, 0xc7b97651866bd192ULL, 0x00a8546d7c558a27ULL,
00019         0x4258b586d5bfbcb4ULL, 0x5e1c3d753d46d260ULL, 0x1cecdc9e94ace4f3ULL,
00020         0xdbfdfea26e92bf46ULL, 0x990d1f49c77889d5ULL, 0x172f5b3033043ebfULL,
00021         0x55dfbadb9aee082cULL, 0x92ce98e760d05399ULL, 0xd03e790cc93a650aULL,
00022         0xaa478900b1228e31ULL, 0xe8b768eb18c8b8a2ULL, 0x2fa64ad7e2f6e317ULL,
00023         0x6d56ab3c4b1cd584ULL, 0xe374ef45bf6062eeULL, 0xa1840eae168a547dULL,
00024         0x66952c92ecb40fc8ULL, 0x2465cd79455e395bULL, 0x3821458aada7578fULL,
00025         0x7ad1a461044d611cULL, 0xbdc0865dfe733aa9ULL, 0xff3067b657990c3aULL,
00026         0x711223cfa3e5bb50ULL, 0x33e2c2240a0f8dc3ULL, 0xf4f3e018f031d676ULL,
00027         0xb60301f359dbe0e5ULL, 0xda050215ea6c212fULL, 0x98f5e3fe438617bcULL,
00028         0x5fe4c1c2b9b84c09ULL, 0x1d14202910527a9aULL, 0x93366450e42ecdf0ULL,
00029         0xd1c685bb4dc4fb63ULL, 0x16d7a787b7faa0d6ULL, 0x5427466c1e109645ULL,
00030         0x4863ce9ff6e9f891ULL, 0x0a932f745f03ce02ULL, 0xcd820d48a53d95b7ULL,
00031         0x8f72eca30cd7a324ULL, 0x0150a8daf8ab144eULL, 0x43a04931514122ddULL,
00032         0x84b16b0dab7f7968ULL, 0xc6418ae602954ffbULL, 0xbc387aea7a8da4c0ULL,
00033         0xfec89b01d3679253ULL, 0x39d9b93d2959c9e6ULL, 0x7b2958d680b3ff75ULL,
00034         0xf50b1caf74cf481fULL, 0xb7fbfd44dd257e8cULL, 0x70eadf78271b2539ULL,
00035         0x321a3e938ef113aaULL, 0x2e5eb66066087d7eULL, 0x6cae578bcfe24bedULL,
00036         0xabbf75b735dc1058ULL, 0xe94f945c9c3626cbULL, 0x676dd025684a91a1ULL,
00037         0x259d31cec1a0a732ULL, 0xe28c13f23b9efc87ULL, 0xa07cf2199274ca14ULL,
00038         0x167ff3eacbaf2af1ULL, 0x548f120162451c62ULL, 0x939e303d987b47d7ULL,
00039         0xd16ed1d631917144ULL, 0x5f4c95afc5edc62eULL, 0x1dbc74446c07f0bdULL,
00040         0xdaad56789639ab08ULL, 0x985db7933fd39d9bULL, 0x84193f60d72af34fULL,
00041         0xc6e9de8b7ec0c5dcULL, 0x01f8fcb784fe9e69ULL, 0x43081d5c2d14a8faULL,
00042         0xcd2a5925d9681f90ULL, 0x8fdab8ce70822903ULL, 0x48cb9af28abc72b6ULL,
00043         0x0a3b7b1923564425ULL, 0x70428b155b4eaf1eULL, 0x32b26afef2a4998dULL,
00044         0xf5a348c2089ac238ULL, 0xb753a929a170f4abULL, 0x3971ed50550c43c1ULL,
00045         0x7b810cbbfce67552ULL, 0xbc902e8706d82ee7ULL, 0xfe60cf6caf321874ULL,
00046         0xe224479f47cb76a0ULL, 0xa0d4a674ee214033ULL, 0x67c58448141f1b86ULL,
00047         0x253565a3bdf52d15ULL, 0xab1721da49899a7fULL, 0xe9e7c031e063acecULL,
00048         0x2ef6e20d1a5df759ULL, 0x6c0603e6b3b7c1caULL, 0xf6fae5c07d3274cdULL,
00049         0xb40a042bd4d8425eULL, 0x731b26172ee619ebULL, 0x31ebc7fc870c2f78ULL,
00050         0xbfc9838573709812ULL, 0xfd39626eda9aae81ULL, 0x3a28405220a4f534ULL,
00051         0x78d8a1b9894ec3a7ULL, 0x649c294a61b7ad73ULL, 0x266cc8a1c85d9be0ULL,
00052         0xe17dea9d3263c055ULL, 0xa38d0b769b89f6c6ULL, 0x2daf4f0f6ff541acULL,
00053         0x6f5faee4c61f773fULL, 0xa84e8cd83c212c8aULL, 0xeabe6d3395cb1a19ULL,
00054         0x90c79d3fedd3f122ULL, 0xd2377cd44439c7b1ULL, 0x15265ee8be079c04ULL,
00055         0x57d6bf0317edaa97ULL, 0xd9f4fb7ae3911dfdULL, 0x9b041a914a7b2b6eULL,
00056         0x5c1538adb04570dbULL, 0x1ee5d94619af4648ULL, 0x02a151b5f156289cULL,
00057         0x4051b05e58bc1e0fULL, 0x87409262a28245baULL, 0xc5b073890b687329ULL,
00058         0x4b9237f0ff14c443ULL, 0x0962d61b56fef2d0ULL, 0xce73f427acc0a965ULL,
00059         0x8c8315cc052a9ff6ULL, 0x3a80143f5cf17f13ULL, 0x7870f5d4f51b4980ULL,
00060         0xbf61d7e80f251235ULL, 0xfd913603a6cf24a6ULL, 0x73b3727a52b393ccULL,
00061         0x31439391fb59a55fULL, 0xf652b1ad0167feeaULL, 0xb4a25046a88dc879ULL,
00062         0xa8e6d8b54074a6adULL, 0xea16395ee99e903eULL, 0x2d071b6213a0cb8bULL,
00063         0x6ff7fa89ba4afd18ULL, 0xe1d5bef04e364a72ULL, 0xa3255f1be7dc7ce1ULL,
00064         0x64347d271de22754ULL, 0x26c49cccb40811c7ULL, 0x5cbd6cc0cc10fafcULL,
00065         0x1e4d8d2b65facc6fULL, 0xd95caf179fc497daULL, 0x9bac4efc362ea149ULL,
00066         0x158e0a85c2521623ULL, 0x577eeb6e6bb820b0ULL, 0x906fc95291867b05ULL,
00067         0xd29f28b9386c4d96ULL, 0xcedba04ad0952342ULL, 0x8c2b41a1797f15d1ULL,
00068         0x4b3a639d83414e64ULL, 0x09ca82762aab78f7ULL, 0x87e8c60fded7cf9dULL,
00069         0xc51827e4773df90eULL, 0x020905d88d03a2bbULL, 0x40f9e43324e99428ULL,
00070         0x2cffe7d5975e55e2ULL, 0x6e0f063e3eb46371ULL, 0xa91e2402c48a38c4ULL,
00071         0xebeec5e96d600e57ULL, 0x65cc8190991cb93dULL, 0x273c607b30f68faeULL,
00072         0xe02d4247cac8d41bULL, 0xa2dda3ac6322e288ULL, 0xbe992b5f8bdb8c5cULL,
00073         0xfc69cab42231bacfULL, 0x3b78e888d80fe17aULL, 0x7988096371e5d7e9ULL,
```

```
00074         0xf7aa4d1a85996083ULL, 0xb55aacf12c735610ULL, 0x724b8ecdd64d0da5ULL,
00075         0x30bb6f267fa73b36ULL, 0x4ac29f2a07bfd00dULL, 0x08327ec1ae55e69eULL,
00076         0xcf235cfd546bbd2bULL, 0x8dd3bd16fd818bb8ULL, 0x03f1f96f09fd3cd2ULL,
00077         0x41011884a0170a41ULL, 0x86103ab85a2951f4ULL, 0xc4e0db53f3c36767ULL,
00078         0xd8a453a01b3a09b3ULL, 0x9a54b24bb2d03f20ULL, 0x5d45907748ee6495ULL,
00079         0x1fb5719ce1045206ULL, 0x919735e51578e56cULL, 0xd367d40ebc92d3ffULL,
00080         0x1476f63246ac884aULL, 0x568617d9ef46bed9ULL, 0xe085162ab69d5e3cULL,
00081         0xa275f7c11f7768afULL, 0x6564d5fde549331aULL, 0x279434164ca30589ULL,
00082         0xa9b6706fb8dfb2e3ULL, 0xeb46918411358470ULL, 0x2c57b3b8eb0bdfc5ULL,
00083         0x6ea7525342e1e956ULL, 0x72e3daa0aa188782ULL, 0x30133b4b03f2b111ULL,
00084         0xf7021977f9cceaa4ULL, 0xb5f2f89c5026dc37ULL, 0x3bd0bce5a45a6b5dULL,
00085         0x79205d0e0db05dceULL, 0xbe317f32f78e067bULL, 0xfcc19ed95e6430e8ULL,
00086         0x86b86ed5267cdbd3ULL, 0xc4488f3e8f96ed40ULL, 0x0359ad0275a8b6f5ULL,
00087         0x41a94ce9dc428066ULL, 0xcf8b0890283e370cULL, 0x8d7be97b81d4019fULL,
00088         0x4a6acb477bea5a2aULL, 0x089a2aacd2006cb9ULL, 0x14dea25f3af9026dULL,
00089         0x562e43b4931334feULL, 0x913f6188692d6f4bULL, 0xd3cf8063c0c759d8ULL,
00090         0x5dedc41a34bbeeb2ULL, 0x1f1d25f19d51d821ULL, 0xd80c07cd676f8394ULL,
00091         0x9afce626ce85b507ULL,
00092     };
00093
00094     inline constexpr auto CRC64Eval(const oxyU8* data, oxySize size) -> oxyU64
00095     {
00096         uint64_t crc{};
00097         uintmax_t i{};
00098         for (; i < size; i++)
00099         {
00100             uint64_t t = (crc » 0x38) ^ data[i];
00101             crc = g_CRC64Table[t & 0xff] ^ (crc « 8);
00102         }
00103         return crc;
00104     }
00105 }; // namespace oxygen
```

7.52 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Math/↩ Random.h File Reference

Namespaces

- namespace oxygen

Functions

- auto oxygen::RandomS32 (oxyS32 minInclusive, oxyS32 maxInclusive) -> oxyS32
- auto oxygen::RandomU32 (oxyU32 minInclusive, oxyU32 maxInclusive) -> oxyU32
- auto oxygen::RandomS64 (oxyS64 minInclusive, oxyS64 maxInclusive) -> oxyS64
- auto oxygen::RandomU64 (oxyU64 minInclusive, oxyU64 maxInclusive) -> oxyU64
- auto oxygen::RandomF32 (oxyF32 minInclusive, oxyF32 maxInclusive) -> oxyF32
- auto oxygen::RandomF64 (oxyF64 minInclusive, oxyF64 maxInclusive) -> oxyF64
- auto oxygen::RandomBool () -> oxyBool

Variables

- std::random_device oxygen::g_randomDevice
- std::mt19937_64 oxygen::g_randomEngine {g_randomDevice()}

7.53 Random.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003
00004
00005 namespace oxygen
00006 {
00007     inline std::random_device g_randomDevice;
00008     inline std::mt19937_64 g_randomEngine{g_randomDevice()};
00009     inline auto RandomS32(oxyS32 minInclusive, oxyS32 maxInclusive) -> oxyS32
00010     {
00011         std::uniform_int_distribution<oxyS32> dist{minInclusive, maxInclusive};
00012         return dist(g_randomEngine);
00013     }
00014     inline auto RandomU32(oxyU32 minInclusive, oxyU32 maxInclusive) -> oxyU32
00015     {
00016         std::uniform_int_distribution<oxyU32> dist{minInclusive, maxInclusive};
00017         return dist(g_randomEngine);
00018     }
00019     inline auto RandomS64(oxyS64 minInclusive, oxyS64 maxInclusive) -> oxyS64
00020     {
00021         std::uniform_int_distribution<oxyS64> dist{minInclusive, maxInclusive};
00022         return dist(g_randomEngine);
00023     }
00024     inline auto RandomU64(oxyU64 minInclusive, oxyU64 maxInclusive) -> oxyU64
00025     {
00026         std::uniform_int_distribution<oxyU64> dist{minInclusive, maxInclusive};
00027         return dist(g_randomEngine);
00028     }
00029     inline auto RandomF32(oxyF32 minInclusive, oxyF32 maxInclusive) -> oxyF32
00030     {
00031         std::uniform_real_distribution<oxyF32> dist{minInclusive, maxInclusive};
00032         return dist(g_randomEngine);
00033     }
00034     inline auto RandomF64(oxyF64 minInclusive, oxyF64 maxInclusive) -> oxyF64
00035     {
00036         std::uniform_real_distribution<oxyF64> dist{minInclusive, maxInclusive};
00037         return dist(g_randomEngine);
00038     }
00039     inline auto RandomBool() -> oxyBool
00040     {
00041         std::bernoulli_distribution dist;
00042         return dist(g_randomEngine);
00043     }
00044
00045 }; // namespace oxygen
```

7.54 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Net/NetSystem.cc File Reference

```
#include "OxygenPCH.h"
#include "NetSystem.h"
#include "GameManager/GameManager.h"
```

Namespaces

- namespace oxygen

7.55 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Net/NetSystem.h File Reference

```
#include "Singleton/Singleton.h"
#include "Platform/Platform.h"
```

Classes

- struct oxygen::NetConnection
- struct oxygen::NetSystem

Namespaces

- namespace oxygen

7.56 NetSystem.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 // this is lazy
00006 #include "Platform/Platform.h"
00007
00008 namespace oxygen
00009 {
00010     struct NetConnection
00011     {
00012         auto WriteData(oxyU16 type, std::span<const oxyU8> data) -> void;
00013
00014         auto GetUniqueID() const -> oxyU64
00015         {
00016             return m_uniqueID;
00017         }
00018
00019       private:
00020         oxyU64 m_uniqueID{};
00021         std::unique_ptr<NetworkAbstraction::NetworkSocket> m_socket{};
00022         std::thread m_receiveThread{};
00023         std::thread m_sendThread{};
00024         struct MessageHeader
00025         {
00026             oxyU16 m_type{};
00027             oxyU16 m_size{};
00028         };
00029
00030         static inline constexpr auto k_queueSize = 1024;
00031         oxyBool m_connected{};
00032
00033         // TODO:
00034         // SPSCQueue is BUSTED
00035         // I'm going to use a mutex for now
00036         // watch this later: https://www.youtube.com/watch?v=K3P_Lmq6pw0
00037
00040         SPSCQueue<std::vector<oxyU8>, k_queueSize>
00041             m_sendQueue{};
00044         SPSCQueue<std::vector<oxyU8>, k_queueSize>
00045             m_receiveQueue{};
00046         // std::queue<std::vector<oxyU8>> m_sendQueue{};
00047         // std::queue<std::vector<oxyU8>> m_receiveQueue{};
00048         // std::mutex m_sendMutex{};
00049         // std::mutex m_receiveMutex{};
00050
00051         friend struct NetSystem;
00052     };
00053
00054     struct NetSystem : SingletonBase<NetSystem>
00055     {
00056         NetSystem();
00057         ~NetSystem();
00058
00059         auto Update(oxyF32 deltaTimeSeconds) -> void;
00060
00061         auto StartHost() -> void;
00062         auto ConnectToHost(const std::string& ip) -> oxyBool;
00063
00064         auto KillConnections() -> void;
00065
00066         auto HostSendToAll(oxyU16 type, const std::span<oxyU8>& data) -> void;
00067         auto HostSendToAllExcept(oxyU64 excludeClientID, oxyU16 type,
```

```
00068                                 const std::span<oxyU8>& data) -> void;
00069
00070         auto CliSendToHost(oxyU16 type, const std::span<oxyU8>& data) -> void;
00071
00072         auto CliDiscoverHosts() -> void;
00073         auto CliGetDiscoveredHosts() const -> std::span<const std::string>
00074         {
00075             return m_discoveredHosts;
00076         }
00077         auto CliIsDiscoveringHosts() const -> oxyBool
00078         {
00079             return m_broadcastDiscoveryRunning;
00080         }
00081
00082         auto GetNewNetObjID() -> oxyObjectID
00083         {
00084             return m_nextNetObjID++;
00085         }
00086
00087         auto IsHost() const -> oxyBool
00088         {
00089             return m_isHost;
00090         }
00091
00092         auto IsClient() const -> oxyBool
00093         {
00094             return m_isClient;
00095         }
00096
00097         static inline constexpr auto k_enginePort = 28672;
00098         static inline constexpr auto k_engineBroadcastPort = 28678;
00099         static inline constexpr auto k_timeBetweenPing = 4.0f;
00100         static inline constexpr auto k_netSessionDefaultMinObjid =
00101             oxyObjectID{0xC0000001};
00102
00103       private:
00104         auto PingAll() -> void;
00105
00106         auto ServerAcceptPeers() -> void;
00107
00108         auto GenNewUniqueID() -> oxyU64;
00109
00110         auto
00111         ReceiveAllFromNetworkSocket(NetworkAbstraction::NetworkSocket& sock)
00112             -> std::optional<std::vector<oxyU8>>;
00113         auto SendAllToNetworkSocket(NetworkAbstraction::NetworkSocket& sock,
00114                                     const std::span<oxyU8>& data) -> oxyBool;
00115
00116         auto ProcessReceivedMessage(NetConnection& conn,
00117                                     std::span<const oxyU8> data) -> void;
00118
00119         auto NetConnectionReceiveThread(NetConnection& connection) -> void;
00120         auto NetConnectionSendThread(NetConnection& connection) -> void;
00121
00122         auto ClientBroadcastSendThread() -> void;
00123         auto ServerBroadcastListenThread() -> void;
00124
00125         // (if client)
00126         // client->server
00127         NetConnection m_clientHostSocket{};
00128         std::unique_ptr<NetworkAbstraction::NetworkSocket>
00129             m_clientBroadcastSendSocket{};
00130         std::thread m_clientBroadcastSendThread{};
00131         oxyBool m_broadcastDiscoveryRunning{};
00132         // broadcast thread out:
00133         std::vector<std::string> m_broadcastDiscoveredHosts{};
00134         std::mutex m_broadcastDiscoveredHostsMutex{};
00135         // for update/main thread:
00136         std::vector<std::string> m_discoveredHosts{};
00137
00138         // (if host)
00139         // server listen socket
00140         std::unique_ptr<NetworkAbstraction::NetworkSocket> m_serverSocket{};
00141         std::unique_ptr<NetworkAbstraction::NetworkSocket>
00142             m_serverBroadcastListenSocket{};
00143         std::thread m_serverBroadcastListenThread{};
00144         std::vector<std::unique_ptr<NetConnection>> m_clients{};
00145
00146         oxyBool m_isHost{};
00147         oxyBool m_isClient{};
00148         oxyBool m_requestShutdown{};
00149
00150         oxyF32 m_timeSinceLastPing{};
00151         oxyObjectID m_nextNetObjID{k_netSessionDefaultMinObjid};
00152     };
00153 }; // namespace oxygen
```

7.57 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/↩ ManagedObject.h File Reference

Classes

- struct oxygen::ManagedObject

Namespaces

- namespace oxygen

7.58 ManagedObject.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     struct ManagedObject : Object
00006     {
00007         OXYGENOBJECT(ManagedObject, Object);
00008
00009         auto GetObjectID() const -> oxyObjectID
00010         {
00011             return m_id;
00012         }
00013
00014         template <typename RefType>
00015         auto GetHardRef() const -> std::shared_ptr<RefType>
00016             requires std::is_base_of_v<ManagedObject, RefType>
00017         {
00018             const auto sptr = m_self.lock();
00019             if (sptr)
00020                 return std::static_pointer_cast<RefType>(std::move(sptr));
00021             return nullptr;
00022         }
00023
00024       private:
00025         oxyObjectID m_id{};
00026         std::weak_ptr<ManagedObject> m_self{};
00027
00028         friend struct ObjectManager;
00029     };
00030 }; // namespace oxygen
```

7.59 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/↩ Object.h File Reference

Classes

- struct oxygen::ObjectDescription
- struct oxygen::Object

Namespaces

- namespace oxygen

Macros

- #define OXYGENOBJECT(Name, Parent)

Functions

- auto oxygen::GetObjectDescriptionMap () -> std::unordered_map< oxyU64, const ObjectDescription * > &

7.59.1 Macro Definition Documentation

7.59.1.1 OXYGENOBJECT

```
#define OXYGENOBJECT(
              Name,
              Parent)
```

7.60 Object.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     struct Object;
00006     struct ObjectDescription
00007     {
00008         const ObjectDescription* m_parent{};
00009         std::string_view m_name{};
00010         oxyU64 m_id{};
00011         oxySize m_size{};
00012         oxySize m_align{};
00013         using constructor_t = Object* (*)(void* p);
00014         constructor_t m_constructor{};
00015     };
00016
00017     inline auto GetObjectDescriptionMap()
00018         -> std::unordered_map<oxyU64, const ObjectDescription*>&
00019     {
00020         static std::unordered_map<oxyU64, const ObjectDescription*> map{};
00021         return map;
00022     }
00023
00024 #define OXYGENOBJECT(Name, Parent)                                          \
00025     Name() = default;                                                       \
00026     ~Name() = default;                                                      \
00027     using SelfType = Name;                                                  \
00028     using Super = Parent;                                                   \
00029     static inline auto GetStaticDescription() -> const ObjectDescription&  \
00030     {                                                                       \
00031         return ObjectInternalDef::g_staticDescriptor;                       \
00032     }                                                                       \
00033     virtual auto GetDescription() const -> const ObjectDescription& override \
00034     {                                                                       \
00035         return GetStaticDescription();                                      \
00036     }                                                                       \
00037                                                                             \
00038   private:                                                                  \
00039     struct ObjectInternalDef                                                \
00040     {                                                                       \
00041         static auto GetStaticDescription() -> const ObjectDescription&     \
00042         {                                                                   \
00043             static_assert(std::is_base_of_v<Object, Parent>,                \
00044                           "Parent must be a subclass of Object");           \
00045             static_assert(std::is_base_of_v<Parent, Name>,                  \
00046                           "Name must be a subclass of Parent");             \
00047             static ObjectDescription desc{                                  \
00048                 .m_parent = &Parent::GetStaticDescription(),                \
00049                 .m_name = #Name,                                            \
```

```
00050                 .m_id = CRC64Eval(reinterpret_cast<const oxyU8*>(#Name),       \
00051                                   sizeof(#Name) - 1),                          \
00052                 .m_size = sizeof(Name),                                        \
00053                 .m_align = alignof(Name),                                      \
00054                 .m_constructor =                                               \
00055                     [](void* p) {                                              \
00056                         return static_cast<Object*>(::new (p) Name{});         \
00057                     },                                                         \
00058             };                                                                 \
00059             const auto mm = reinterpret_cast<Name*>(GetStaticDescription);     \
00060             OXYCHECK((void*)static_cast<Object*>(mm) == (void*)mm);            \
00061             return desc;                                                       \
00062         }                                                                      \
00063         static inline const ObjectDescription& g_staticDescriptor{             \
00064             GetStaticDescription()};                                           \
00065         static inline oxyChar g_staticDescriptorInit = []() {                  \
00066             GetObjectDescriptionMap().emplace(GetStaticDescription().m_id,     \
00067                                               &g_staticDescriptor);            \
00068             return 0;                                                          \
00069         }();                                                                   \
00070     };                                                                         \
00071                                                                                \
00072     public:
00073
00074       struct Object
00075       {
00076           using SelfType = Object;
00077           using Super = Object;
00078
00079           Object() = default;
00080           virtual ~Object() = default;
00081
00082           static inline auto GetStaticDescription() -> const ObjectDescription&
00083           {
00084               return ObjectInternalDef::g_staticDescriptor;
00085           }
00086           virtual auto GetDescription() const -> const ObjectDescription&
00087           {
00088               return GetStaticDescription();
00089           }
00090
00091           auto IsA(const ObjectDescription& desc) const -> bool
00092           {
00093               auto* p = &GetDescription();
00094               while (p)
00095               {
00096                   if (p->m_id == desc.m_id)
00097                       return true;
00098                   p = p->m_parent;
00099               }
00100               return false;
00101           }
00102           template <typename T>
00103           auto IsA() const -> bool
00104               requires std::is_base_of_v<Object, T>
00105           {
00106               return IsA(T::GetStaticDescription());
00107           }
00108           template <typename T>
00109           auto Cast() -> T*
00110               requires std::is_base_of_v<Object, T>
00111           {
00112               return IsA<T>() ? static_cast<T*>(this) : nullptr;
00113           }
00114           template <typename T>
00115           auto Cast() const -> const T*
00116               requires std::is_base_of_v<Object, T>
00117           {
00118               return IsA<T>() ? static_cast<const T*>(this) : nullptr;
00119           }
00120
00121         private:
00122           struct ObjectInternalDef
00123           {
00124               static auto GetStaticDescription() -> const ObjectDescription&
00125               {
00126                   static ObjectDescription desc{
00127                       .m_parent = nullptr,
00128                       .m_name = "Object",
00129                       .m_id = CRC64Eval(reinterpret_cast<const oxyU8*>("Object"),
00130                                         sizeof("Object") - 1),
00131                       .m_size = sizeof(Object),
00132                       .m_align = alignof(Object),
00133                       .m_constructor = [](void* p) { return ::new (p) Object{}; },
00134                   };
00135                   return desc;
00136               }
```

```
00137             static inline const ObjectDescription& g_staticDescriptor{
00138                 GetStaticDescription()};
00139             static inline oxyChar g_staticDescriptorInit = []() {
00140                 GetObjectDescriptionMap().emplace(GetStaticDescription().m_id,
00141                                                   &g_staticDescriptor);
00142                 return 1;
00143             }();
00144         };
00145     };
00146
00147 }; // namespace oxygen
```

7.61 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/↩ ObjectManager.cc File Reference

```
#include "OxygenPCH.h"
#include "ObjectManager.h"
```

Namespaces

- namespace oxygen

7.62 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Object/↩ ObjectManager.h File Reference

```
#include "Singleton/Singleton.h"
```

Classes

- struct oxygen::ObjectManager

Namespaces

- namespace oxygen

7.63 ObjectManager.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 namespace oxygen
00006 {
00007     struct ObjectManager : SingletonBase<ObjectManager>
00008     {
00009         ~ObjectManager();
00010
00011         template <typename T>
00012         auto NewObject(oxyObjectID id = 0) -> T*
00013             requires std::is_base_of_v<Object, T> &&
```

```
00014                   std::is_same_v<typename T::SelfType, T>;
00015         auto NewObject(const ObjectDescription& desc,
00016                        oxyObjectID id = 0) -> Object*;
00017         auto DeleteObject(Object* obj, oxyObjectID id = 0) -> void;
00018
00019         template <typename T>
00020         auto CreateManagedObject(oxyObjectID id = 0) -> std::shared_ptr<T>
00021             requires std::is_base_of_v<ManagedObject, T> &&
00022                      std::is_same_v<typename T::SelfType, T>;
00023         auto CreateManagedObject(const ObjectDescription& desc,
00024                                  oxyObjectID id = 0)
00025             -> std::shared_ptr<ManagedObject>;
00026
00027         template <typename T>
00028         auto GetObjectPtr(oxyObjectID id) const -> T*
00029             requires std::is_base_of_v<Object, T>;
00030         auto GetObjectPtr(oxyObjectID id) const -> Object*;
00031
00032         auto GetObjectID(Object* obj) const -> oxyObjectID;
00033
00034         template <typename T>
00035         auto GetManagedRef(oxyObjectID id) const -> std::shared_ptr<T>
00036             requires std::is_base_of_v<ManagedObject, T>;
00037         auto
00038         GetManagedRef(oxyObjectID id) const -> std::shared_ptr<ManagedObject>;
00039
00040       private:
00041         oxyObjectID m_nextDynamicID{0x40000001};
00042         std::unordered_map<oxyObjectID, Object*> m_objects{};
00043         std::unordered_map<Object*, oxyObjectID> m_objectIDs{};
00044
00045         oxyObjectID m_nextManagedID{0x80000001};
00046         std::unordered_map<oxyObjectID, std::weak_ptr<ManagedObject»
00047             m_managedObjects{};
00048         struct ManagedObjectDeleter
00049         {
00050             auto operator()(ManagedObject* obj) -> void;
00051         };
00052     };
00053
00054     template <typename T>
00055     auto ObjectManager::NewObject(oxyObjectID id) -> T*
00056         requires std::is_base_of_v<Object, T> &&
00057                  std::is_same_v<typename T::SelfType, T>
00058     {
00059         if (!id)
00060             id = ++m_nextDynamicID;
00061         else
00062         {
00063             const auto it = m_objects.find(id);
00064             if (it != m_objects.end())
00065                 return static_cast<T*>(it->second);
00066         }
00067         const auto storage =
00068             ::operator new[](sizeof(T), std::align_val_t{alignof(T)});
00069         const auto obj = new (storage) T{};
00070         OXYCHECK(storage == obj);
00071         m_objects.emplace(id, obj);
00072         m_objectIDs.emplace(obj, id);
00073         return obj;
00074     }
00075
00076     template <typename T>
00077     auto
00078     ObjectManager::CreateManagedObject(oxyObjectID id) -> std::shared_ptr<T>
00079         requires std::is_base_of_v<ManagedObject, T> &&
00080                  std::is_same_v<typename T::SelfType, T>
00081     {
00082         if (!id)
00083             id = ++m_nextManagedID;
00084         else
00085         {
00086             const auto it = m_managedObjects.find(id);
00087             if (it != m_managedObjects.end())
00088             {
00089                 auto sptr = it->second.lock();
00090                 if (sptr)
00091                     return std::static_pointer_cast<T>(std::move(sptr));
00092                 return nullptr;
00093             }
00094         }
00095         const auto ptr = NewObject<T>(id);
00096         auto sptr = std::shared_ptr<T>(ptr, ManagedObjectDeleter{});
00097         ptr->m_id = id;
00098         ptr->m_self = sptr;
00099         m_managedObjects.emplace(id, sptr);
00100         return sptr;
```

```
00101     }
00102
00103     template <typename T>
00104     auto ObjectManager::GetObjectPtr(oxyObjectID id) const -> T*
00105         requires std::is_base_of_v<Object, T>
00106     {
00107         return static_cast<T*>(GetObjectPtr(id));
00108     }
00109
00110     template <typename T>
00111     auto
00112     ObjectManager::GetManagedRef(oxyObjectID id) const -> std::shared_ptr<T>
00113         requires std::is_base_of_v<ManagedObject, T>
00114     {
00115         auto ref = std::static_pointer_cast<T>(GetManagedRef(id));
00116         if (ref)
00117         {
00118             if (!ref->IsA<T>())
00119                 ref = {};
00120         }
00121         return ref;
00122     }
00123 }; // namespace oxygen
```

7.64 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/OxygenPCH.cc File Reference

```
#include "OxygenPCH.h"
```

7.65 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/OxygenPCH.h File Reference

```
#include "Platform/InternalPCHBase.h"
```

7.66 OxygenPCH.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #if defined(_WIN32) && defined(_WIN64) && defined(_MSC_VER)
00004 #include "Platform/PlatformWin64/PrecompiledHeaders/PCH.h"
00005 #else
00006 #error "Unsupported platform"
00007 #endif
00008
00009 #include "Platform/InternalPCHBase.h"
```

7.67 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/OxygenTypes.h File Reference

```
#include <cstdint>
#include "Math/Defs.h"
#include "Math/Hash.h"
#include "Math/Random.h"
#include "Containers/SPSCQueue.h"
```

Classes

- struct oxygen::NonCopyable

  A non-copyable type to be inherited from. Expresses clearly that a type cannot be copied.
- struct oxygen::CallbackList< TRet, TArgs >

Namespaces

- namespace oxygen

Typedefs

- using oxyU8 = std::uint8_t
- using oxyU16 = std::uint16_t
- using oxyU32 = std::uint32_t
- using oxyU64 = std::uint64_t
- using oxyS8 = std::int8_t
- using oxyS16 = std::int16_t
- using oxyS32 = std::int32_t
- using oxyS64 = std::int64_t
- using oxyF32 = float
- using oxyF64 = double
- using oxyBool = bool
- using oxyChar = char
- using oxyWChar = wchar_t
- using oxySize = size_t
- using oxySSize = ptrdiff_t
- using oxyObjectID = oxyU64

Enumerations

- enum oxygen::ControllerButton : oxyU8 {
  oxygen::ControllerButton_LeftThumb , oxygen::ControllerButton_RightThumb , oxygen::ControllerButton_LeftShoulder , oxygen::ControllerButton_RightShoulder ,
  oxygen::ControllerButton_South , oxygen::ControllerButton_East , oxygen::ControllerButton_West , oxygen::ControllerButton_North ,
  oxygen::ControllerButton_StartSelect , oxygen::ControllerButton_BackShare , oxygen::ControllerButton_DPadUp , oxygen::ControllerButton_DPadDown ,
  oxygen::ControllerButton_DPadLeft , oxygen::ControllerButton_DPadRight , oxygen::ControllerButton_Count
  }
- enum oxygen::ControllerAxis : oxyU8 {
  oxygen::ControllerAxis_LeftThumbX , oxygen::ControllerAxis_LeftThumbY , oxygen::ControllerAxis_RightThumbX , oxygen::ControllerAxis_RightThumbY ,
  oxygen::ControllerAxis_LeftTrigger , oxygen::ControllerAxis_RightTrigger , oxygen::ControllerAxis_Count }
- enum oxygen::MouseButton : oxyU8 {
  oxygen::MouseButton_Left , oxygen::MouseButton_Right , oxygen::MouseButton_Middle , oxygen::MouseButton_X1 ,
  oxygen::MouseButton_X2 , oxygen::MouseButton_Count }

- enum oxygen::KeyboardButton : oxyU8 {
  oxygen::KeyboardButton_A , oxygen::KeyboardButton_B , oxygen::KeyboardButton_C , oxygen::KeyboardButton_D
  ,
  oxygen::KeyboardButton_E , oxygen::KeyboardButton_F , oxygen::KeyboardButton_G , oxygen::KeyboardButton_H
  ,
  oxygen::KeyboardButton_I , oxygen::KeyboardButton_J , oxygen::KeyboardButton_K , oxygen::KeyboardButton_L
  ,
  oxygen::KeyboardButton_M , oxygen::KeyboardButton_N , oxygen::KeyboardButton_O , oxygen::KeyboardButton_P
  ,
  oxygen::KeyboardButton_Q , oxygen::KeyboardButton_R , oxygen::KeyboardButton_S , oxygen::KeyboardButton_T
  ,
  oxygen::KeyboardButton_U , oxygen::KeyboardButton_V , oxygen::KeyboardButton_W , oxygen::KeyboardButton_X
  ,
  oxygen::KeyboardButton_Y , oxygen::KeyboardButton_Z , oxygen::KeyboardButton_0 , oxygen::KeyboardButton_1
  ,
  oxygen::KeyboardButton_2 , oxygen::KeyboardButton_3 , oxygen::KeyboardButton_4 , oxygen::KeyboardButton_5
  ,
  oxygen::KeyboardButton_6 , oxygen::KeyboardButton_7 , oxygen::KeyboardButton_8 , oxygen::KeyboardButton_9
  ,
  oxygen::KeyboardButton_F1 , oxygen::KeyboardButton_F2 , oxygen::KeyboardButton_F3 , oxygen::KeyboardButton_F4
  ,
  oxygen::KeyboardButton_F5 , oxygen::KeyboardButton_F6 , oxygen::KeyboardButton_F7 , oxygen::KeyboardButton_F8
  ,
  oxygen::KeyboardButton_F9 , oxygen::KeyboardButton_F10 , oxygen::KeyboardButton_F11 , oxygen::KeyboardButton_F12
  ,
  oxygen::KeyboardButton_F13 , oxygen::KeyboardButton_F14 , oxygen::KeyboardButton_F15 , oxygen::KeyboardButton_F16
  ,
  oxygen::KeyboardButton_F17 , oxygen::KeyboardButton_F18 , oxygen::KeyboardButton_F19 , oxygen::KeyboardButton_F20
  ,
  oxygen::KeyboardButton_F21 , oxygen::KeyboardButton_F22 , oxygen::KeyboardButton_F23 , oxygen::KeyboardButton_F24
  ,
  oxygen::KeyboardButton_NumPad0 , oxygen::KeyboardButton_NumPad1 , oxygen::KeyboardButton_NumPad2
  , oxygen::KeyboardButton_NumPad3 ,
  oxygen::KeyboardButton_NumPad4 , oxygen::KeyboardButton_NumPad5 , oxygen::KeyboardButton_NumPad6
  , oxygen::KeyboardButton_NumPad7 ,
  oxygen::KeyboardButton_NumPad8 , oxygen::KeyboardButton_NumPad9 , oxygen::KeyboardButton_NumPadDecimal
  , oxygen::KeyboardButton_NumPadEnter ,
  oxygen::KeyboardButton_NumPadAdd , oxygen::KeyboardButton_NumPadSubtract , oxygen::KeyboardButton_NumPadMultipl
  , oxygen::KeyboardButton_NumPadDivide ,
  oxygen::KeyboardButton_NumPadLock , oxygen::KeyboardButton_Left , oxygen::KeyboardButton_Right ,
  oxygen::KeyboardButton_Up ,
  oxygen::KeyboardButton_Down , oxygen::KeyboardButton_Home , oxygen::KeyboardButton_End ,
  oxygen::KeyboardButton_PageUp ,
  oxygen::KeyboardButton_PageDown , oxygen::KeyboardButton_Insert , oxygen::KeyboardButton_Delete ,
  oxygen::KeyboardButton_Pause ,
  oxygen::KeyboardButton_PrintScreen , oxygen::KeyboardButton_ScrollLock , oxygen::KeyboardButton_Escape
  , oxygen::KeyboardButton_Backtick ,
  oxygen::KeyboardButton_Tab , oxygen::KeyboardButton_CapsLock , oxygen::KeyboardButton_LeftShift ,
  oxygen::KeyboardButton_LeftControl ,
  oxygen::KeyboardButton_LeftWindows , oxygen::KeyboardButton_LeftAlt , oxygen::KeyboardButton_Space ,
  oxygen::KeyboardButton_RightAlt ,
  oxygen::KeyboardButton_RightFunction , oxygen::KeyboardButton_RightMenu , oxygen::KeyboardButton_RightControl
  , oxygen::KeyboardButton_RightShift ,
  oxygen::KeyboardButton_Enter , oxygen::KeyboardButton_Backspace , oxygen::KeyboardButton_Comma ,
  oxygen::KeyboardButton_Period ,
  oxygen::KeyboardButton_Slash , oxygen::KeyboardButton_Semicolon , oxygen::KeyboardButton_Apostrophe
  , oxygen::KeyboardButton_LeftBracket ,
  oxygen::KeyboardButton_RightBracket , oxygen::KeyboardButton_Backslash , oxygen::KeyboardButton_Hyphen

, oxygen::KeyboardButton_Equals ,
oxygen::KeyboardButton_Count }

- enum oxygen::CollisionHull : oxyU8 {
oxygen::CollisionHull_None = 0xFF , oxygen::CollisionHull_Point = 0 , oxygen::CollisionHull_Player ,
oxygen::CollisionHull_PlayerCrouched ,
oxygen::CollisionHull_Grenade }
- enum oxygen::CollisionResponseType : oxyU8 { oxygen::CollisionResponseType_None , oxygen::CollisionResponseType_Slide
, oxygen::CollisionResponseType_Bounce }
- enum oxygen::EntitySummonType : oxyU8 { oxygen::EntitySummonType_Player , oxygen::EntitySummonType_Count
}
- enum oxygen::AnimationHash : oxyU32 {
oxygen::AnimationHash_Idle = 0x7c161a2b , oxygen::AnimationHash_RunForward = 0x947ec374 ,
oxygen::AnimationHash_RunBackward = 0x144ff8d , oxygen::AnimationHash_Dying = 0x12c8a4ff ,
oxygen::AnimationHash_Throw = 0x8e526e33 }
- enum oxygen::EntitySpawnType : oxyU8 {
oxygen::EntitySpawnType_Player , oxygen::EntitySpawnType_Golfclub , oxygen::EntitySpawnType_GolfclubLauncher
, oxygen::EntitySpawnType_Golfball ,
oxygen::EntitySpawntype_GolfballLauncher , oxygen::EntitySpawnType_Count }
- enum oxygen::NetProtoMsgType : oxyU16 {
oxygen::NetProtoMsgType_AnyPing = 0 , oxygen::NetProtoMsgType_SrvWelcome , oxygen::NetProtoMsgType_SrvChangeLev
, oxygen::NetProtoMsgType_SrvSetLocalPlayer ,
oxygen::NetProtoMsgType_SrvEntitySpawn , oxygen::NetProtoMsgType_SrvEntityDestroy , oxygen::NetProtoMsgType_SrvEnt
, oxygen::NetProtoMsgType_SrvHealthComponentChange ,
oxygen::NetProtoMsgType_SrvPawnPickupWeapon , oxygen::NetProtoMsgType_SrvPawnDropWeapon ,
oxygen::NetProtoMsgType_CliLocalPlayerEntityMove , oxygen::NetProtoMsgType_CliPawnDropWeapon ,
oxygen::NetProtoMsgType_CliLocalPlayerFireWeapon }
- enum oxygen::PickupType : oxyU8 { oxygen::PickupType_Health , oxygen::PickupType_Ammo ,
oxygen::PickupType_Weapon , oxygen::PickupType_Count }
- enum oxygen::WeaponFireType : oxyU8 { oxygen::WeaponFireType_Bullets , oxygen::WeaponFireType_GolfClub
, oxygen::WeaponFireType_GolfBall , oxygen::WeaponFireType_Count }
- enum oxygen::HealthState : oxyU8 { oxygen::HealthState_Alive , oxygen::HealthState_Invulnerable ,
oxygen::HealthState_Dead }
- enum oxygen::DamageType : oxyU8 {
oxygen::DamageType_None , oxygen::DamageType_Explosive , oxygen::DamageType_Bullet , oxygen::DamageType_Melee
,
oxygen::DamageType_FallDamage , oxygen::DamageType_Count }

Variables

- constexpr oxyVec3 oxygen::k_collisionHullMins [ ]
- constexpr oxyVec3 oxygen::k_collisionHullMaxs [ ]

7.67.1 Typedef Documentation

7.67.1.1 oxyBool

```
using oxyBool = bool
```

7.67.1.2 oxyChar

```
using oxyChar = char
```

7.67.1.3 oxyF32

```
using oxyF32 = float
```

7.67.1.4 oxyF64

```
using oxyF64 = double
```

7.67.1.5 oxyObjectID

```
using oxyObjectID = oxyU64
```

7.67.1.6 oxyS16

```
using oxyS16 = std::int16_t
```

7.67.1.7 oxyS32

```
using oxyS32 = std::int32_t
```

7.67.1.8 oxyS64

```
using oxyS64 = std::int64_t
```

7.67.1.9 oxyS8

```
using oxyS8 = std::int8_t
```

7.67.1.10 oxySize

```
using oxySize = size_t
```

7.67.1.11 oxySSize

```
using oxySSize = ptrdiff_t
```

7.67.1.12 oxyU16

```
using oxyU16 = std::uint16_t
```

7.67.1.13 oxyU32

```
using oxyU32 = std::uint32_t
```

7.67.1.14 oxyU64

```
using oxyU64 = std::uint64_t
```

7.67.1.15 oxyU8

```
using oxyU8 = std::uint8_t
```

7.67.1.16 oxyWChar

```
using oxyWChar = wchar_t
```

7.68 OxygenTypes.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include <cstdint>
00004
00005 using oxyU8 = std::uint8_t;
00006 using oxyU16 = std::uint16_t;
00007 using oxyU32 = std::uint32_t;
00008 using oxyU64 = std::uint64_t;
00009 using oxyS8 = std::int8_t;
00010 using oxyS16 = std::int16_t;
00011 using oxyS32 = std::int32_t;
00012 using oxyS64 = std::int64_t;
00013 using oxyF32 = float;
00014 using oxyF64 = double;
00015
00016 using oxyBool = bool;
00017
00018 using oxyChar = char;
00019 using oxyWChar = wchar_t;
00020
00021 using oxySize = size_t;
00022 using oxySSize = ptrdiff_t;
00023
00024 using oxyObjectID = oxyU64;
00025
00026 #include "Math/Defs.h"
00027 #include "Math/Hash.h"
00028 #include "Math/Random.h"
00029
00030 #include "Containers/SPSCQueue.h"
00031
00032 namespace oxygen
00033 {
00039     struct NonCopyable
00040     {
00041         NonCopyable() = default;
00042         NonCopyable(const NonCopyable&) = delete;
00043         NonCopyable& operator=(const NonCopyable&) = delete;
00044     };
00045
00046     enum ControllerButton : oxyU8
00047     {
00048         ControllerButton_LeftThumb,
00049         ControllerButton_RightThumb,
00050         ControllerButton_LeftShoulder,
00051         ControllerButton_RightShoulder,
```

```
00052         ControllerButton_South,
00053         ControllerButton_East,
00054         ControllerButton_West,
00055         ControllerButton_North,
00056         ControllerButton_StartSelect,
00057         ControllerButton_BackShare,
00058         ControllerButton_DPadUp,
00059         ControllerButton_DPadDown,
00060         ControllerButton_DPadLeft,
00061         ControllerButton_DPadRight,
00062         ControllerButton_Count
00063     };
00064
00065     enum ControllerAxis : oxyU8
00066     {
00067         ControllerAxis_LeftThumbX,
00068         ControllerAxis_LeftThumbY,
00069         ControllerAxis_RightThumbX,
00070         ControllerAxis_RightThumbY,
00071         ControllerAxis_LeftTrigger,
00072         ControllerAxis_RightTrigger,
00073         ControllerAxis_Count
00074     };
00075
00076     enum MouseButton : oxyU8
00077     {
00078         MouseButton_Left,
00079         MouseButton_Right,
00080         MouseButton_Middle,
00081         MouseButton_X1,
00082         MouseButton_X2,
00083         MouseButton_Count
00084     };
00085
00086     enum KeyboardButton : oxyU8
00087     {
00088         // A-Z
00089         KeyboardButton_A,
00090         KeyboardButton_B,
00091         KeyboardButton_C,
00092         KeyboardButton_D,
00093         KeyboardButton_E,
00094         KeyboardButton_F,
00095         KeyboardButton_G,
00096         KeyboardButton_H,
00097         KeyboardButton_I,
00098         KeyboardButton_J,
00099         KeyboardButton_K,
00100         KeyboardButton_L,
00101         KeyboardButton_M,
00102         KeyboardButton_N,
00103         KeyboardButton_O,
00104         KeyboardButton_P,
00105         KeyboardButton_Q,
00106         KeyboardButton_R,
00107         KeyboardButton_S,
00108         KeyboardButton_T,
00109         KeyboardButton_U,
00110         KeyboardButton_V,
00111         KeyboardButton_W,
00112         KeyboardButton_X,
00113         KeyboardButton_Y,
00114         KeyboardButton_Z,
00115
00116         // 0-9
00117         KeyboardButton_0,
00118         KeyboardButton_1,
00119         KeyboardButton_2,
00120         KeyboardButton_3,
00121         KeyboardButton_4,
00122         KeyboardButton_5,
00123         KeyboardButton_6,
00124         KeyboardButton_7,
00125         KeyboardButton_8,
00126         KeyboardButton_9,
00127
00128         // F1-F24
00129         KeyboardButton_F1,
00130         KeyboardButton_F2,
00131         KeyboardButton_F3,
00132         KeyboardButton_F4,
00133         KeyboardButton_F5,
00134         KeyboardButton_F6,
00135         KeyboardButton_F7,
00136         KeyboardButton_F8,
00137         KeyboardButton_F9,
00138         KeyboardButton_F10,
```

```
00139         KeyboardButton_F11,
00140         KeyboardButton_F12,
00141         KeyboardButton_F13,
00142         KeyboardButton_F14,
00143         KeyboardButton_F15,
00144         KeyboardButton_F16,
00145         KeyboardButton_F17,
00146         KeyboardButton_F18,
00147         KeyboardButton_F19,
00148         KeyboardButton_F20,
00149         KeyboardButton_F21,
00150         KeyboardButton_F22,
00151         KeyboardButton_F23,
00152         KeyboardButton_F24,
00153
00154         // Numpad 0-9
00155         KeyboardButton_NumPad0,
00156         KeyboardButton_NumPad1,
00157         KeyboardButton_NumPad2,
00158         KeyboardButton_NumPad3,
00159         KeyboardButton_NumPad4,
00160         KeyboardButton_NumPad5,
00161         KeyboardButton_NumPad6,
00162         KeyboardButton_NumPad7,
00163         KeyboardButton_NumPad8,
00164         KeyboardButton_NumPad9,
00165
00166         // Numpad etc
00167         KeyboardButton_NumPadDecimal,
00168         KeyboardButton_NumPadEnter,
00169         KeyboardButton_NumPadAdd,
00170         KeyboardButton_NumPadSubtract,
00171         KeyboardButton_NumPadMultiply,
00172         KeyboardButton_NumPadDivide,
00173         KeyboardButton_NumPadLock,
00174
00175         // Arrow keys
00176         KeyboardButton_Left,
00177         KeyboardButton_Right,
00178         KeyboardButton_Up,
00179         KeyboardButton_Down,
00180
00181         // Home/End, Page Up/Down, Insert/Delete
00182         KeyboardButton_Home,
00183         KeyboardButton_End,
00184         KeyboardButton_PageUp,
00185         KeyboardButton_PageDown,
00186         KeyboardButton_Insert,
00187         KeyboardButton_Delete,
00188
00189         // Pause/Break, Print Screen, Scroll Lock
00190         KeyboardButton_Pause,
00191         KeyboardButton_PrintScreen,
00192         KeyboardButton_ScrollLock,
00193
00194         // Escape, Backtick, Tab, Caps Lock, Left Shift, Left Control, Left
00195         // Windows, Left Alt
00196         KeyboardButton_Escape,
00197         KeyboardButton_Backtick,
00198         KeyboardButton_Tab,
00199         KeyboardButton_CapsLock,
00200         KeyboardButton_LeftShift,
00201         KeyboardButton_LeftControl,
00202         KeyboardButton_LeftWindows,
00203         KeyboardButton_LeftAlt,
00204
00205         // Space
00206         KeyboardButton_Space,
00207
00208         // Right Alt, Right Function, Right Menu, Right Control, Right Shift,
00209         // Enter, Backspace
00210         KeyboardButton_RightAlt,
00211         KeyboardButton_RightFunction,
00212         KeyboardButton_RightMenu,
00213         KeyboardButton_RightControl,
00214         KeyboardButton_RightShift,
00215         KeyboardButton_Enter,
00216         KeyboardButton_Backspace,
00217
00218         // Comma, Period, Slash, Semicolon, Apostrophe, Left Bracket, Right
00219         // Bracket, Backslash, Hyphen, Equals
00220         KeyboardButton_Comma,
00221         KeyboardButton_Period,
00222         KeyboardButton_Slash,
00223         KeyboardButton_Semicolon,
00224         KeyboardButton_Apostrophe,
00225         KeyboardButton_LeftBracket,
```

```
00226         KeyboardButton_RightBracket,
00227         KeyboardButton_Backslash,
00228         KeyboardButton_Hyphen,
00229         KeyboardButton_Equals,
00230 
00231         KeyboardButton_Count
00232     };
00233 
00234     enum CollisionHull : oxyU8
00235     {
00236         CollisionHull_None = 0xFF,
00237         CollisionHull_Point = 0,
00238         CollisionHull_Player,
00239         CollisionHull_PlayerCrouched,
00240         CollisionHull_Grenade,
00241     };
00242 
00243     inline constexpr oxyVec3 k_collisionHullMins[] = {
00244         oxyVec3{0.f, 0.f, 0.f},         // CollisionHull_Point
00245         oxyVec3{-24.f, -24.f, -48.f}, // CollisionHull_Player
00246         oxyVec3{-24.f, -24.f, -48.f}, // CollisionHull_PlayerCrouched
00247         oxyVec3{-12.f, -12.f, -12.f},       // CollisionHull_Grenade
00248     };
00249     inline constexpr oxyVec3 k_collisionHullMaxs[] = {
00250         oxyVec3{0.f, 0.f, 0.f},      // CollisionHull_Point
00251         oxyVec3{24.f, 24.f, 48.f}, // CollisionHull_Player
00252         oxyVec3{24.f, 24.f, 48.f}, // CollisionHull_PlayerCrouched
00253         oxyVec3{12.f, 12.f, 12.f},       // CollisionHull_Grenade
00254     };
00255 
00256     enum CollisionResponseType : oxyU8
00257     {
00258         CollisionResponseType_None,
00259         CollisionResponseType_Slide,
00260         CollisionResponseType_Bounce,
00261     };
00262 
00263     enum EntitySummonType : oxyU8
00264     {
00265         EntitySummonType_Player,
00266         EntitySummonType_Count
00267     };
00268 
00269     enum AnimationHash : oxyU32
00270     {
00271         AnimationHash_Idle = 0x7c161a2b,
00272         AnimationHash_RunForward = 0x947ec374,
00273         AnimationHash_RunBackward = 0x144ff8d,
00274         AnimationHash_Dying = 0x12c8a4ff,
00275         AnimationHash_Throw = 0x8e526e33,
00276     };
00277 
00278     enum EntitySpawnType : oxyU8
00279     {
00280         EntitySpawnType_Player,
00281         EntitySpawnType_Golfclub,
00282         EntitySpawnType_GolfclubLauncher,
00283         EntitySpawnType_Golfball,
00284         EntitySpawntype_GolfballLauncher,
00285         EntitySpawnType_Count
00286     };
00287 
00288     enum NetProtoMsgType : oxyU16
00289     {
00290         NetProtoMsgType_AnyPing = 0,
00291 
00292         // srv->cl
00293         // oxyU64: unique client id (only the client and host know this, not
00294         // other peers)
00295         NetProtoMsgType_SrvWelcome,
00296 
00297         // [utf8 string]: world name
00298         NetProtoMsgType_SrvChangeLevel,
00299 
00300         // oxyObjectID: entity id to possess
00301         NetProtoMsgType_SrvSetLocalPlayer,
00302         // oxyU16: EntitySpawnType
00303         // oxyVec3: position
00304         // oxyQuat: rotation
00305         // oxyU16: count of object ids
00306         // for each object id:
00307         // oxyObjectID: entity (if idx 0), or component id for spawning
00308         NetProtoMsgType_SrvEntitySpawn,
00309 
00310         // oxyObjectID: entity id
00311         NetProtoMsgType_SrvEntityDestroy,
00312 
```

```
00313         // oxyU16: count of entities in msg
00314         // for each entity:
00315         // oxyObjectID: entity id
00316         // oxyVec3: position
00317         // oxyQuat: rotation
00318         NetProtoMsgType_SrvEntityTransformRepl,
00319 
00320         // oxyObjectID: health component id
00321         // oxyS32: health
00322         // oxyS32: max health
00323         // oxyU8: health state
00324         NetProtoMsgType_SrvHealhComponentChange,
00325 
00326         // oxyObjectID: pawn component id
00327         // oxyObjectID: weapon component id
00328         NetProtoMsgType_SrvPawnPickupWeapon,
00329 
00330         // oxyObjectID: pawn component id
00331         NetProtoMsgType_SrvPawnDropWeapon,
00332 
00333         // client->srv
00334         // oxyVec3: position
00335         // oxyQuat: rotation
00336         NetProtoMsgType_CliLocalPlayerEntityMove,
00337 
00338         // oxyObjectID: pawn component id
00339         NetProtoMsgType_CliPawnDropWeapon,
00340 
00341         // oxyVec3: euler
00342         // oxyVec3: position
00343         // oxyU8: righthanded
00344         NetProtoMsgType_CliLocalPlayerFireWeapon,
00345 
00346     };
00347 
00348     enum PickupType : oxyU8
00349     {
00350         PickupType_Health,
00351         PickupType_Ammo,
00352         PickupType_Weapon,
00353         PickupType_Count
00354     };
00355 
00356     enum WeaponFireType : oxyU8
00357     {
00358         WeaponFireType_Bullets,
00359         WeaponFireType_GolfClub,
00360         WeaponFireType_GolfBall,
00361         WeaponFireType_Count
00362     };
00363 
00364     enum HealthState : oxyU8
00365     {
00366         HealthState_Alive,
00367         HealthState_Invulnerable,
00368         HealthState_Dead,
00369     };
00370 
00371     enum DamageType : oxyU8
00372     {
00373         DamageType_None,
00374         DamageType_Explosive,
00375         DamageType_Bullet,
00376         DamageType_Melee,
00377         DamageType_FallDamage,
00378         DamageType_Count,
00379     };
00380 
00381     template <typename TRet, typename... TArgs> struct CallbackList
00382     {
00383         using fnptr_type = TRet (*)(void*, TArgs...);
00384 
00385         auto AddCallback(std::weak_ptr<void> obj, fnptr_type fn) -> void
00386         {
00387             m_callbacks.emplace_back(std::move(obj), fn);
00388         }
00389 
00390         template <typename TFun>
00391         auto IterateCallbacks(TFun&& fun, TArgs... args)
00392 
00393             -> void
00394             requires(not std::same_as<void, TRet>)
00395         {
00396             for (auto it = m_callbacks.begin(); it != m_callbacks.end();)
00397             {
00398                 auto& [obj, fn] = *it;
00399                 if (auto sp = obj.lock())
```

```
00400                 {
00401                     fun(fn(sp, std::forward<TArgs>(args)...));
00402                     ++it;
00403                 }
00404                 else
00405                 {
00406                     it = m_callbacks.erase(it);
00407                 }
00408             }
00409         }
00410 
00411         template <typename TFun>
00412         auto IterateCallbacks(TFun&& fun, TArgs... args)
00413 
00414             -> void
00415         {
00416             for (auto it = m_callbacks.begin(); it != m_callbacks.end();)
00417             {
00418                 auto& [obj, fn] = *it;
00419                 if (auto sp = obj.lock())
00420                 {
00421                     fn(sp.get(), std::forward<TArgs>(args)...);
00422                     fun();
00423                     ++it;
00424                 }
00425                 else
00426                 {
00427                     it = m_callbacks.erase(it);
00428                 }
00429             }
00430         }
00431 
00432       private:
00433         std::vector<std::pair<std::weak_ptr<void>, fnptr_type» m_callbacks;
00434     };
00435 
00436 }; // namespace oxygen
```

7.69 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Platform/InternalPCHBase.h File Reference

```
#include <memory>
#include <memory_resource>
#include <utility>
#include <execution>
#include <thread>
#include <future>
#include <atomic>
#include <algorithm>
#include <ranges>
#include <mutex>
#include <optional>
#include <variant>
#include <array>
#include <string>
#include <vector>
#include <unordered_map>
#include <bitset>
#include <string_view>
#include <span>
#include <format>
#include <cmath>
#include <random>
#include "OxygenTypes.h"
#include "Object/Object.h"
#include "Object/ManagedObject.h"
#include "Object/ObjectManager.h"
```

7.70 InternalPCHBase.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 // memory
00004 // #include <new> // transitive include from <memory>
00005 #include <memory>
00006 #include <memory_resource>
00007
00008 // util
00009 #include <utility>
00010
00011 // execution
00012 #include <execution>
00013
00014 // threading
00015 #include <thread>
00016 #include <future>
00017 #include <atomic>
00018
00019 // algorithm/ranges
00020 #include <algorithm>
00021 #include <ranges>
00022
00023 // threading
00024 #include <thread>
00025 #include <future>
00026 #include <atomic>
00027 #include <mutex> // deeply ashamed of this one
00028
00029 // containers
00030 #include <optional>
00031 #include <variant>
00032 #include <array>
00033 #include <string>
00034 #include <vector>
00035 #include <unordered_map>
00036 #include <bitset>
00037
00038 // spans
00039 #include <string_view>
00040 #include <span>
00041
00042 // formatting
00043 #include <format>
00044
00045 // math
00046 #include <cmath>
00047
00048 // random
00049 #include <random>
00050
00051 // engine types
00052 #include "OxygenTypes.h"
00053 #include "Object/Object.h"
00054 #include "Object/ManagedObject.h"
00055 #include "Object/ObjectManager.h"
```

7.71 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Platform/↩ Platform.h File Reference

7.72 Platform.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #pragma once
00003
00004 namespace oxygen
00005 {
00006     auto GetExecutableDirectory() -> std::string_view;
00007
00008     auto GetLaunchArguments() -> std::span<const std::string>;
00009
```

```
00010     auto LogMessage(const char* str) -> void;
00011
00012     auto ReadFileContents(std::string_view absolutePath) -> std::vector<oxyU8>;
00013
00014     struct FileMap : NonCopyable
00015     {
00016         auto GetMap() const -> void*
00017         {
00018             return m_data;
00019         }
00020         auto GetSize() const -> oxySize
00021         {
00022             return m_size;
00023         }
00024         auto ValidateRange(const void* ptr, oxySize sz) const -> bool
00025         {
00026             return reinterpret_cast<const oxyU8*>(ptr) >=
00027                        reinterpret_cast<const oxyU8*>(m_data) &&
00028                    (reinterpret_cast<const oxyU8*>(ptr) + sz) <=
00029                        (reinterpret_cast<const oxyU8*>(m_data) + m_size) &&
00030                    sz && m_data;
00031         }
00032
00033       protected:
00034         FileMap() = default;
00035         ~FileMap() = default;
00036         void* m_data{};
00037         oxySize m_size{};
00038     };
00039     struct InternalFileMapDeleter
00040     {
00041         auto operator()(FileMap* ptr) const -> void;
00042     };
00043     using UniqueFileMap = std::unique_ptr<FileMap, InternalFileMapDeleter>;
00044     auto CreateFileMap(std::string_view path, oxyBool write = false,
00045                        oxySize requestSize = 0) -> UniqueFileMap;
00046
00047     namespace GraphicsAbstraction
00048     {
00049         auto GetWindowSize(oxyS32& width, oxyS32& height) -> void;
00050
00051         struct Texture
00052         {
00053             oxyU32 m_width;
00054             oxyU32 m_height;
00055             void* m_internalPlatformHandle;
00056         };
00057         auto
00058         LoadTexture(const char* absolutePath) -> std::shared_ptr<const Texture>;
00059
00060         struct TexturedQuad
00061         {
00062             // NDC, -1 to +1
00063             oxyVec2 m_vertices[4];
00064             // OpenGL style uvs
00065             oxyVec2 m_textureCoords[4];
00066             // RGB
00067             oxyVec3 m_colour;
00068             // Sample texture
00069             const Texture* m_texture;
00070         };
00071         auto DrawTexturedQuad(const TexturedQuad& quad) -> void;
00072     }; // namespace GraphicsAbstraction
00073
00074     namespace AudioAbstraction
00075     {
00076
00077     };
00078
00079     namespace InputAbstraction
00080     {
00081         auto HideAndLockCursor(oxyBool lock) -> void;
00082         auto GetMousePosition(oxyF32& x, oxyF32& y) -> void;
00083         auto GetMouseStates(std::bitset<MouseButton_Count>& buttons) -> void;
00084         auto GetKeyStates(std::bitset<KeyboardButton_Count>& keys) -> void;
00085         auto GetControllerConnected(int index) -> oxyBool;
00086         auto GetControllerStates(
00087             int index, std::bitset<ControllerButton_Count>& buttons) -> void;
00088         auto GetControllerAxisStates(
00089             int index, std::span<oxyF32, ControllerAxis_Count> axes) -> void;
00090
00091     }; // namespace InputAbstraction
00092
00093     namespace NetworkAbstraction
00094     {
00095         struct NetworkSocket
00096         {
```

```
00097             ~NetworkSocket();
00098
00099             auto Send(const void* data, oxySize size) -> oxySSize;
00100             auto Receive(void* data, oxySize size) -> oxySSize;
00101
00102             auto Accept() -> std::unique_ptr<NetworkSocket>;
00103
00104             auto BroadcastMessage(const void* data, oxySize size) -> std::vector<std::string>;
00105             auto RespondToBroadcasts(const void* data, oxySize size) -> void;
00106
00107           private:
00108             OXYSOCKETDESCRIPTORTYPE
00109             m_descriptor{}; // might need to be changed, unix uses signed
00110             oxyU32 m_address{};
00111             oxyU16 m_port{};
00112
00113             friend auto ConnectToHost(const char* host, oxyU16 port)
00114                 -> std::unique_ptr<NetworkSocket>;
00115             friend auto
00116             HostServer(oxyU16 port) -> std::unique_ptr<NetworkSocket>;
00117             friend auto CreateBroadcastSendSocket(oxyU16 port)
00118                 -> std::unique_ptr<NetworkSocket>;
00119             friend auto CreateBroadcastListenSocket(oxyU16 port)
00120                 -> std::unique_ptr<NetworkSocket>;
00121         };
00122         auto ConnectToHost(const char* host,
00123                            oxyU16 port) -> std::unique_ptr<NetworkSocket>;
00124         auto HostServer(oxyU16 port) -> std::unique_ptr<NetworkSocket>;
00125         auto
00126         CreateBroadcastSendSocket(oxyU16 port) -> std::unique_ptr<NetworkSocket>;
00127         auto CreateBroadcastListenSocket(oxyU16 port)
00128             -> std::unique_ptr<NetworkSocket>;
00129
00130     }; // namespace NetworkAbstraction
00131
00132 }; // namespace oxygen
```

7.73 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Platform/PlatformWin64/Platform.cc File Reference

```
#include "OxygenPCH.h"
#include "Platform/Platform.h"
#include "Singleton/EngineSingletons.h"
#include "App/app.h"
#include "PrivateMembers.h"
```

Classes

- struct oxygen::InternalFileMapWinX64

Namespaces

- namespace oxygen
- namespace oxygen::GraphicsAbstraction
- namespace oxygen::AudioAbstraction
- namespace oxygen::InputAbstraction
- namespace oxygen::NetworkAbstraction

Macros

- #define SETBTNDOWN(idx, appParam)
- #define SETKEYDOWN(idx, appParam)

Functions

- auto oxygen::GetExecutableDirectory () -> std::string_view
- auto oxygen::GetLaunchArguments () -> std::span< const std::string >
- auto oxygen::LogMessage (const char *str) -> void
- auto oxygen::ReadFileContents (std::string_view absolutePath) -> std::vector< oxyU8 >
- auto oxygen::CreateFileMap (std::string_view path, oxyBool write, oxySize requestSize) -> UniqueFileMap
- auto oxygen::Win64PlatformInit () -> void
- auto oxygen::Win64PlatformRender () -> void
- auto oxygen::Win64PlatformUpdate (float deltaTimeSeconds) -> void
- auto oxygen::Win64PlatformShutdown () -> void
- auto oxygen::GraphicsAbstraction::GetWindowSize (oxyS32 &width, oxyS32 &height) -> void
- auto oxygen::GraphicsAbstraction::LoadTexture (const char *absolutePath) -> std::shared_ptr< const Texture >
- auto oxygen::GraphicsAbstraction::DrawTexturedQuad (const TexturedQuad &quad) -> void
- auto oxygen::InputAbstraction::IsForeground () -> oxyBool
- auto oxygen::InputAbstraction::HideAndLockCursor (oxyBool lock) -> void
- auto oxygen::InputAbstraction::GetMousePosition (oxyF32 &x, oxyF32 &y) -> void
- auto oxygen::InputAbstraction::GetMouseStates (std::bitset< MouseButton_Count > &buttons) -> void
- auto oxygen::InputAbstraction::GetKeyStates (std::bitset< KeyboardButton_Count > &keys) -> void
- auto oxygen::InputAbstraction::GetControllerConnected (int index) -> oxyBool
- auto oxygen::InputAbstraction::GetControllerStates (int index, std::bitset< ControllerButton_Count > &buttons) -> void
- auto oxygen::InputAbstraction::GetControllerAxisStates (int index, std::span< oxyF32, ControllerAxis_Count > axes) -> void
- auto oxygen::NetworkAbstraction::ConnectToHost (const char *host, oxyU16 port) -> std::unique_ptr< NetworkSocket >
- auto oxygen::NetworkAbstraction::HostServer (oxyU16 port) -> std::unique_ptr< NetworkSocket >
- auto oxygen::NetworkAbstraction::CreateBroadcastSendSocket (oxyU16 port) -> std::unique_ptr< NetworkSocket >
- auto oxygen::NetworkAbstraction::CreateBroadcastListenSocket (oxyU16 port) -> std::unique_ptr< NetworkSocket >

Variables

- int WINDOW_WIDTH
- int WINDOW_HEIGHT
- HWND MAIN_WINDOW_HANDLE

7.73.1 Macro Definition Documentation

7.73.1.1 SETBTNDOWN

```
#define SETBTNDOWN(
              idx,
              appParam)
```

Value:

```
buttons[idx] = App::IsKeyPressed(appParam);
```

7.73.1.2 SETKEYDOWN

```
#define SETKEYDOWN(
              idx,
              appParam)
```

Value:

```
keys[idx] = App::IsKeyPressed(appParam);
```

7.73.2 Variable Documentation

7.73.2.1 MAIN_WINDOW_HANDLE

```
HWND MAIN_WINDOW_HANDLE  [extern]
```

7.73.2.2 WINDOW_HEIGHT

```
int WINDOW_HEIGHT  [extern]
```

7.73.2.3 WINDOW_WIDTH

```
int WINDOW_WIDTH  [extern]
```

7.74 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Platform/↩ PlatformWin64/PrecompiledHeaders/PCH.h File Reference

```
#include <intrin.h>
```

Macros

- #define OXYGEN_PLATFORM_WIN64
- #define OXYENDIANLITTLE
- #define OXYSOCKETDESCRIPTORTYPE oxyU64
- #define OXYDEBUGBREAK()
- #define OXYCHECK(x)
- #define OXYVERIFY(x)

7.74.1 Macro Definition Documentation

7.74.1.1 OXYCHECK

```
#define OXYCHECK(
              x)
```

Value:

```
    do                                                                  \
    {                                                                   \
    } while (0)
```

7.74.1.2 OXYDEBUGBREAK

```
#define OXYDEBUGBREAK()
```

Value:

```
__writeeflags(__readeflags() | 0x100)
```

7.74.1.3 OXYENDIANLITTLE

```
#define OXYENDIANLITTLE
```

7.74.1.4 OXYGEN_PLATFORM_WIN64

```
#define OXYGEN_PLATFORM_WIN64
```

7.74.1.5 OXYSOCKETDESCRIPTORTYPE

```
#define OXYSOCKETDESCRIPTORTYPE oxyU64
```

7.74.1.6 OXYVERIFY

```
#define OXYVERIFY(
               x)
```

Value:

```
    do                                                                  \
    {                                                                   \
        x;                                                              \
    } while (0)
```

7.75 PCH.h

Go to the documentation of this file.

```
00001 #define OXYGEN_PLATFORM_WIN64
00002
00003 #include <intrin.h>
00004
00005 #define OXYENDIANLITTLE
00006
00007 #define OXYSOCKETDESCRIPTORTYPE oxyU64
00008
00009 //
00010 // engine macros
00011 //
00012 // OXYDEBUGBREAK
00013 // Switches the thread into single-step mode, allowing you to debug code
00014 // line-by-line and continue execution if needed
00015 #define OXYDEBUGBREAK() __writeeflags(__readeflags() | 0x100)
00016
00017 // OXYCHECK asserts in debug builds, but does not evaluate or execute in release
00018 #ifdef OXYBUILDDEBUG
00019 #define OXYCHECK(x)                                                 \
00020     do                                                              \
00021     {                                                               \
00022         if (!(x))                                                   \
00023         {                                                           \
00024             OXYDEBUGBREAK();                                        \
```

```
00025         }                                                                   \
00026     } while (0)
00027 #else
00028 #define OXYCHECK(x)                                                         \
00029     do                                                                      \
00030     {                                                                       \
00031     } while (0)
00032 #endif
00033 // OXYVERIFY
00034 // OXYVERIFY asserts only in debug builds, but still evaluates and executes in
00035 // release
00036 #ifdef OXYBUILDDEBUG
00037 #define OXYVERIFY(x)                                                        \
00038     do                                                                      \
00039     {                                                                       \
00040         if (!(x))                                                           \
00041         {                                                                   \
00042             OXYDEBUGBREAK();                                                \
00043         }                                                                   \
00044     } while (0)
00045 #else
00046 #define OXYVERIFY(x)                                                        \
00047     do                                                                      \
00048     {                                                                       \
00049         x;                                                                  \
00050     } while (0)
00051 #endif
```

7.76 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Platform/PlatformWin64/PrivateMembers.h File Reference

```
#include "App/app.h"
```

Namespaces

- namespace oxygen
- namespace oxygen::GraphicsAbstraction
- namespace oxygen::InputAbstraction

7.77 PrivateMembers.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #include "App/app.h"
00003
00004 //
      https://web.archive.org/web/20120401132446/http://bloglitb.blogspot.com/2011/12/access-to-private-members-safer.html
00005 // This technique was invented by litb in 2010, long before
00006 // templates were supercharged in C++17. Inspired by this method, I have
00007 // reimplemented it using more modern C++ features.
00008 //
00009 // https://www.youtube.com/watch?v=SmlLdd1Q2V8
00010 /*
00011 14.7.2p8 The usual access checking rules do not apply to names used to specify
00012 explicit instantiations. [Note: In particular, the template arguments and names
00013 used in the function declarator (including parameter types, return types and
00014 exception specifications) may be private types or objects which would normally
00015 not be accessible and the template may be a member template or member function
00016 which would not normally be accessible.]
00017 */
00018 namespace oxygen
00019 {
00020     namespace
00021     {
00022
00023         // When explicitly instantiated, this assigns a
00024         // non-type-template-parameter value to a reference during global
```

```
00025         // initialization
00026         template <auto& Where, auto What>
00027             requires std::convertible_to<decltype(Where), decltype(What)>
00028         struct NTTPAssigner
00029         {
00030             static inline decltype(auto) s_assignmentReturnResult{Where = What};
00031         };
00032
00033         // int PrivateClass::*g_privateClassBPointer{};
00034         // template struct NTTPAssigner<g_privateClassBPointer,
00035         // &PrivateClass::b>;
00036         // Effectively, this is roughly equivalent to:
00037         // int PrivateClass::*g_privateClassBPointer = &PrivateClass::b;
00038     }; // namespace
00039     namespace GraphicsAbstraction
00040     {
00041         // CSimpleSprite:
00042         // float m_xpos;
00043         static inline float CSimpleSprite::*g_CSimpleSpriteMemberPointerMXPos{};
00044         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMXPos,
00045                                      &CSimpleSprite::m_xpos>;
00046         // float m_ypos;
00047         static inline float CSimpleSprite::*g_CSimpleSpriteMemberPointerMYPos{};
00048         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMYPos,
00049                                      &CSimpleSprite::m_ypos>;
00050         // int m_texWidth;
00051         static inline int CSimpleSprite::*
00052             g_CSimpleSpriteMemberPointerMTexWidth{};
00053         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMTexWidth,
00054                                      &CSimpleSprite::m_texWidth>;
00055
00056         // int m_texHeight;
00057         static inline int CSimpleSprite::*
00058             g_CSimpleSpriteMemberPointerMTexHeight{};
00059         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMTexHeight,
00060                                      &CSimpleSprite::m_texHeight>;
00061
00062         // float m_angle;
00063         static inline float CSimpleSprite::*
00064             g_CSimpleSpriteMemberPointerMAngle{};
00065         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMAngle,
00066                                      &CSimpleSprite::m_angle>;
00067         // float m_scale;
00068         static inline float CSimpleSprite::*
00069             g_CSimpleSpriteMemberPointerMScale{};
00070         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMScale,
00071                                      &CSimpleSprite::m_scale>;
00072         // float m_points[8];
00073         static inline float (
00074             CSimpleSprite::*g_CSimpleSpriteMemberPointerMPoints)[8]{};
00075         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMPoints,
00076                                      &CSimpleSprite::m_points>;
00077         // float m_uvcoords[8];
00078         static inline float (
00079             CSimpleSprite::*g_CSimpleSpriteMemberPointerMUVCoords)[8]{};
00080         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMUVCoords,
00081                                      &CSimpleSprite::m_uvcoords>;
00082         // float m_red;
00083         static inline float CSimpleSprite::*g_CSimpleSpriteMemberPointerMRed{};
00084         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMRed,
00085                                      &CSimpleSprite::m_red>;
00086         // float m_green;
00087         static inline float CSimpleSprite::*
00088             g_CSimpleSpriteMemberPointerMGreen{};
00089         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMGreen,
00090                                      &CSimpleSprite::m_green>;
00091         // float m_blue;
00092         static inline float CSimpleSprite::*g_CSimpleSpriteMemberPointerMBlue{};
00093         template struct NTTPAssigner<g_CSimpleSpriteMemberPointerMBlue,
00094                                      &CSimpleSprite::m_blue>;
00095     }; // namespace GraphicsAbstraction
00096
00097     namespace InputAbstraction
00098     {
00099         // bool m_bConnected = false;
00100         static inline bool CController::*g_CControllerMemberPointerMConnected{};
00101         template struct NTTPAssigner<g_CControllerMemberPointerMConnected,
00102                                      &CController::m_bConnected>;
00103     }; // namespace InputAbstraction
00104 };     // namespace oxygen
```

7.78 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Resources/AnimatedMeshResource.h File Reference

Classes

- struct oxygen::AnimationInfo
- struct oxygen::AnimatedMeshResource

Namespaces

- namespace oxygen

7.79 AnimatedMeshResource.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     struct AnimationInfo
00006     {
00007         std::vector<std::vector<oxyVec3» m_frames;
00008     };
00009     struct AnimatedMeshResource
00010     {
00011         std::shared_ptr<const struct StaticMeshResource> m_rootPose;
00012         std::unordered_map<oxyU32, AnimationInfo> m_animations;
00013     };
00014 };
```

7.80 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Resources/ResourceManager.cc File Reference

```
#include "OxygenPCH.h"
#include "ResourceManager.h"
#include "Platform/Platform.h"
#include "StaticMeshResource.h"
#include "AnimatedMeshResource.h"
```

Namespaces

- namespace oxygen

7.81 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Resources/ResourceManager.h File Reference

```
#include "Singleton/Singleton.h"
```

Classes

- struct oxygen::ResourceManager

Namespaces

- namespace oxygen

7.82 ResourceManager.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 namespace oxygen
00006 {
00007     struct StaticMeshResource;
00008     struct AnimatedMeshResource;
00009
00010     struct ResourceManager : SingletonBase<ResourceManager>
00011     {
00012         auto LoadStaticMesh(std::string_view name)
00013             -> std::shared_ptr<const StaticMeshResource>;
00014         auto LoadAnimatedMesh(std::string_view name)
00015             -> std::shared_ptr<const AnimatedMeshResource>;
00016
00017       private:
00018         std::unordered_map<std::size_t, std::weak_ptr<const StaticMeshResource»
00019             m_staticMeshes;
00020         std::unordered_map<std::size_t,
00021                            std::weak_ptr<const AnimatedMeshResource»
00022             m_AnimatedMeshes;
00023     };
00024 }; // namespace oxygen
```

7.83 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/↩ Resources/StaticMeshResource.h File Reference

Classes

- struct oxygen::StaticMeshVertex
- struct oxygen::StaticMeshTri
- struct oxygen::StaticMeshPointDef
- struct oxygen::StaticMeshResource

Namespaces

- namespace oxygen

7.84 StaticMeshResource.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     struct StaticMeshVertex
00006     {
00007         oxyVec3 m_position;
00008         oxyVec2 m_uv;
00009     };
00010
00011     struct StaticMeshTri
00012     {
00013         StaticMeshVertex m_vertices[3];
00014     };
00015
00016     struct StaticMeshPointDef
00017     {
00018         oxyU32 m_hash;
00019         oxyVec3 m_position;
00020     };
00021
00022     struct StaticMeshResource
00023     {
00024         std::vector<StaticMeshPointDef> m_points;
00025         std::vector<StaticMeshTri> m_tris;
00026         std::string m_texname;
00027     };
00028 };
```

7.85 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Singleton/↩ EngineSingletons.h File Reference

```
#include "Singleton.h"
#include "Object/ObjectManager.h"
#include "Input/InputManager.h"
#include "Gfx/GfxRenderer.h"
#include "Resources/ResourceManager.h"
#include "UI/UIManager.h"
#include "Net/NetSystem.h"
#include "GameManager/GameManager.h"
```

Classes

- struct oxygen::InternalEngineSingletonsOrder

Namespaces

- namespace oxygen

Typedefs

- using oxygen::EngineSingletons = SingletonHolder<InternalEngineSingletonsOrder>

7.86 EngineSingletons.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton.h"
00004 #include "Object/ObjectManager.h"
00005 #include "Input/InputManager.h"
00006 #include "Gfx/GfxRenderer.h"
00007 #include "Resources/ResourceManager.h"
00008 #include "UI/UIManager.h"
00009 #include "Net/NetSystem.h"
00010 #include "GameManager/GameManager.h"
00011
00012 namespace oxygen
00013 {
00014     struct InternalEngineSingletonsOrder
00015     {
00016         SingletonInstance<ObjectManager> m_objectManagerInstance{};
00017         SingletonInstance<InputManager> m_inputManagerInstance{};
00018         SingletonInstance<GfxRenderer> m_gfxRendererInstance{};
00019         SingletonInstance<ResourceManager> m_resourceManagerInstance{};
00020         SingletonInstance<UIManager> m_uiManagerInstance{};
00021         SingletonInstance<NetSystem> m_netSystemInstance{};
00022         SingletonInstance<GameManager> m_gameManagerInstance{};
00023     };
00024
00025     using EngineSingletons = SingletonHolder<InternalEngineSingletonsOrder>;
00026 }; // namespace oxygen
```

7.87 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/Singleton/Singleton.h File Reference

Classes

- struct oxygen::SingletonInstance< T >

  Singleton instance, should only be in an object templated to SingletonHolder. Contains the actual storage buffer for the type T.
- struct oxygen::SingletonBase< CRTPType >

  Singleton base class. All singletons should inherit from this class. Uses the curiously recurring template pattern and provides the GetInstance() static method to access the singleton.
- struct oxygen::SingletonHolder< SingletonsStruct >

  Storage of a group of singletons. Used to explicitly construct and destruct singletons.

Namespaces

- namespace oxygen

7.88 Singleton.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00010     template <typename T> struct SingletonInstance : NonCopyable
00011     {
00012         template <typename... Args> SingletonInstance(Args&&... args)
00013         {
00014             if (!s_inLifeTime)
00015             {
```

```
00016                 new (s_singletonBuffer) T(std::forward<Args>(args)...);
00017                 s_inLifeTime = true;
00018             }
00019         }
00020         ~SingletonInstance()
00021         {
00022             if (s_inLifeTime)
00023             {
00024                 reinterpret_cast<T*>(s_singletonBuffer)->~T();
00025                 s_inLifeTime = false;
00026             }
00027         }
00028         static auto GetInstance() -> T&
00029         {
00030             return *reinterpret_cast<T*>(s_singletonBuffer);
00031         }
00032
00033       private:
00034         static inline bool s_inLifeTime{};
00035         static inline alignas(T) unsigned char s_singletonBuffer[sizeof(T)]{};
00036     };
00037
00044     template <typename CRTPType> struct SingletonBase : NonCopyable
00045     {
00046         SingletonBase()
00047         {
00048             OXYCHECK(!s_instance);
00049             s_instance = static_cast<CRTPType*>(this);
00050         }
00051         ~SingletonBase()
00052         {
00053             s_instance = nullptr;
00054         }
00055         static auto GetInstance() -> CRTPType&
00056         {
00057             return SingletonInstance<CRTPType>::GetInstance();
00058         }
00059
00060       private:
00061         // Purely for debugging:
00062         static inline CRTPType* s_instance{};
00063     };
00064
00070     template <typename SingletonsStruct> struct SingletonHolder : NonCopyable
00071     {
00072         static auto Construct() -> void
00073         {
00074             if (!m_storage.has_value())
00075             {
00076                 m_storage.emplace();
00077             }
00078         }
00079         static auto Destruct() -> void
00080         {
00081             m_storage.reset();
00082         }
00083
00084       private:
00085         // The global optional will have its own atexit destructor guaranteed.
00086         // Thus, all singletons are also guaranteed to be destructed.
00087         static inline std::optional<SingletonsStruct> m_storage{};
00088     };
00089 } // namespace oxygen
```

7.89 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/UI/↩ UIManager.cc File Reference

```
#include "OxygenPCH.h"
#include "UIManager.h"
#include "Gfx/GfxRenderer.h"
#include "Input/InputManager.h"
#include "Net/NetSystem.h"
#include "GameManager/GameManager.h"
```

Namespaces

- namespace oxygen

7.90 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/UI/↩ UIManager.h File Reference

```
#include "Singleton/Singleton.h"
```

Classes

- struct oxygen::UIManager

Namespaces

- namespace oxygen

7.91 UIManager.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "Singleton/Singleton.h"
00004
00005 namespace oxygen
00006 {
00007     struct UIManager : SingletonBase<UIManager>
00008     {
00009         auto Render() const -> void;
00010         auto Update() -> void;
00011
00012         auto DisplayPopup(std::string message) -> void;
00013
00014       private:
00015         enum SelectableUIElements
00016         {
00017             SelectableUIElements_None,
00018             SelectableUIElements_Back,
00019             SelectableUIElements_Host,
00020             SelectableUIElements_Join,
00021             SelectableUIElements_Options,
00022             SelectableUIElements_Quit,
00023
00024             SelectableUIElements_HostListSelection,
00025
00026             SelectableUIElements_Count,
00027         };
00028
00029         oxyS32 m_width{};
00030         oxyS32 m_height{};
00031
00032         oxyBool m_mainMenuOpen{true};
00033         std::vector<std::string> m_popups{};
00034
00035             int m_hoverHostSelectionIndex{-1};
00036         SelectableUIElements m_hoverItem{SelectableUIElements_None};
00037         std::bitset<SelectableUIElements_Count> m_selectedItems{};
00038
00039         auto MousePosNDC() const -> oxyVec2;
00040
00041         auto DrawMainMenu() const -> void;
00042
00043         auto RefreshHostList() -> void;
```

```
00044
00045         auto MainMenuItemSelected() const -> oxyBool
00046         {
00047             return m_selectedItems.test(SelectableUIElements_Host) ||
00048                    m_selectedItems.test(SelectableUIElements_Join) ||
00049                    m_selectedItems.test(SelectableUIElements_Options) ||
00050                    m_selectedItems.test(SelectableUIElements_Quit);
00051         }
00052
00053         auto HoverTextColour(SelectableUIElements expect) const -> oxyVec3;
00054     };
00055 } // namespace oxygen
```

7.92 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.cc File Reference

```
#include "OxygenPCH.h"
#include "BSP.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.93 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/BSP.h File Reference

Classes

- struct oxygen::BSPDefines::Lump
- struct oxygen::BSPDefines::Header
- struct oxygen::BSPDefines::Plane
- struct oxygen::BSPDefines::MipTexLump
- struct oxygen::BSPDefines::MipTex
- struct oxygen::BSPDefines::Vertex
- struct oxygen::BSPDefines::Node
- struct oxygen::BSPDefines::TexInfo
- struct oxygen::BSPDefines::Face
- struct oxygen::BSPDefines::ClipNode
- struct oxygen::BSPDefines::Leaf
- struct oxygen::BSPDefines::Edge
- struct oxygen::BSPDefines::Model
- struct oxygen::BSPWorldData

Namespaces

- namespace oxygen
- namespace oxygen::BSPDefines

Enumerations

- enum oxygen::BSPDefines::Contents {
  oxygen::BSPDefines::Contents_Empty = -1 , oxygen::BSPDefines::Contents_Solid = -2 , oxygen::BSPDefines::Contents_Water = -3 , oxygen::BSPDefines::Contents_Slime = -4 ,
  oxygen::BSPDefines::Contents_Lava = -5 , oxygen::BSPDefines::Contents_Sky = -6 , oxygen::BSPDefines::Contents_Origin = -7 , oxygen::BSPDefines::Contents_Clip = -8 ,
  oxygen::BSPDefines::Contents_Current0 = -9 , oxygen::BSPDefines::Contents_Current90 = -10 ,
  oxygen::BSPDefines::Contents_Current180 = -11 , oxygen::BSPDefines::Contents_Current270 = -12 ,
  oxygen::BSPDefines::Contents_CurrentUp = -13 , oxygen::BSPDefines::Contents_CurrentDown = -14 ,
  oxygen::BSPDefines::Contents_Translucent = -15 }
- enum oxygen::BSPDefines::LumpIndex {
  oxygen::BSPDefines::LumpIndex_Entities = 0 , oxygen::BSPDefines::LumpIndex_Planes , oxygen::BSPDefines::LumpIndex_Te , oxygen::BSPDefines::LumpIndex_Vertexes ,
  oxygen::BSPDefines::LumpIndex_Visibility , oxygen::BSPDefines::LumpIndex_Nodes , oxygen::BSPDefines::LumpIndex_TexIn , oxygen::BSPDefines::LumpIndex_Faces ,
  oxygen::BSPDefines::LumpIndex_Lighting , oxygen::BSPDefines::LumpIndex_ClipNodes , oxygen::BSPDefines::LumpIndex_L , oxygen::BSPDefines::LumpIndex_MarkSurfaces ,
  oxygen::BSPDefines::LumpIndex_Edges , oxygen::BSPDefines::LumpIndex_SurfEdges , oxygen::BSPDefines::LumpIndex_Mo , oxygen::BSPDefines::LumpIndex_Count }
- enum oxygen::BSPDefines::PlaneType {
  oxygen::BSPDefines::Plane_X = 0 , oxygen::BSPDefines::Plane_Y , oxygen::BSPDefines::Plane_Z ,
  oxygen::BSPDefines::Plane_AnyX ,
  oxygen::BSPDefines::Plane_AnyY , oxygen::BSPDefines::Plane_AnyZ }

Variables

- constexpr auto oxygen::BSPDefines::k_BSPVersion = oxyS32{30}
- constexpr auto oxygen::BSPDefines::k_ToolVersion = oxyS32{2}
- constexpr auto oxygen::BSPDefines::k_MaxMapHulls = oxySize{4}
- constexpr auto oxygen::BSPDefines::k_MaxMapModels = oxySize{400}
- constexpr auto oxygen::BSPDefines::k_MaxMapBrushes = oxySize{4096}
- constexpr auto oxygen::BSPDefines::k_MaxMapEntityString = oxySize{128 * 1024}
- constexpr auto oxygen::BSPDefines::k_MaxMapPlanes = oxySize{32767}
- constexpr auto oxygen::BSPDefines::k_MaxMapNodes = oxySize{32767}
- constexpr auto oxygen::BSPDefines::k_MaxMapClipNodes = oxySize{32767}
- constexpr auto oxygen::BSPDefines::k_MaxMapLeafs = oxySize{8192}
- constexpr auto oxygen::BSPDefines::k_MaxMapVertices = oxySize{65535}
- constexpr auto oxygen::BSPDefines::k_MaxMapFaces = oxySize{65535}
- constexpr auto oxygen::BSPDefines::k_MaxMapMarkSurfaces = oxySize{65535}
- constexpr auto oxygen::BSPDefines::k_MaxMapTexInfo = oxySize{8192}
- constexpr auto oxygen::BSPDefines::k_MaxMapEdges = oxySize{256000}
- constexpr auto oxygen::BSPDefines::k_MaxMapSurfEdges = oxySize{512000}
- constexpr auto oxygen::BSPDefines::k_MaxMapTextures = oxySize{512}
- constexpr auto oxygen::BSPDefines::k_MaxMapMipTex = oxySize{0x200000}
- constexpr auto oxygen::BSPDefines::k_MaxMapLighting = oxySize{0x200000}
- constexpr auto oxygen::BSPDefines::k_MaxMapVis = oxySize{0x200000}
- constexpr auto oxygen::BSPDefines::k_MaxMapPortals = oxySize{65536}
- constexpr auto oxygen::BSPDefines::k_NumMipLevels = 4
- constexpr auto oxygen::BSPDefines::k_TexSpecial
- constexpr auto oxygen::BSPDefines::k_MaxLightMaps = 4
- constexpr auto oxygen::BSPDefines::k_NumAmbients = 4

7.94 BSP.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     namespace BSPDefines
00006     {
00007         inline constexpr auto k_BSPVersion = oxyS32{30};
00008         inline constexpr auto k_ToolVersion = oxyS32{2};
00009
00010         inline constexpr auto k_MaxMapHulls = oxySize{4};
00011         inline constexpr auto k_MaxMapModels = oxySize{400};
00012         inline constexpr auto k_MaxMapBrushes = oxySize{4096};
00013         inline constexpr auto k_MaxMapEntityString = oxySize{128 * 1024};
00014         inline constexpr auto k_MaxMapPlanes = oxySize{32767};
00015         inline constexpr auto k_MaxMapNodes = oxySize{32767};
00016         inline constexpr auto k_MaxMapClipNodes = oxySize{32767};
00017         inline constexpr auto k_MaxMapLeafs = oxySize{8192};
00018         inline constexpr auto k_MaxMapVertices = oxySize{65535};
00019         inline constexpr auto k_MaxMapFaces = oxySize{65535};
00020         inline constexpr auto k_MaxMapMarkSurfaces = oxySize{65535};
00021         inline constexpr auto k_MaxMapTexInfo = oxySize{8192};
00022         inline constexpr auto k_MaxMapEdges = oxySize{256000};
00023         inline constexpr auto k_MaxMapSurfEdges = oxySize{512000};
00024         inline constexpr auto k_MaxMapTextures = oxySize{512};
00025         inline constexpr auto k_MaxMapMipTex = oxySize{0x200000};
00026         inline constexpr auto k_MaxMapLighting = oxySize{0x200000};
00027         inline constexpr auto k_MaxMapVis = oxySize{0x200000};
00028         inline constexpr auto k_MaxMapPortals = oxySize{65536};
00029
00030         inline constexpr auto k_NumMipLevels = 4;
00031         inline constexpr auto k_TexSpecial =
00032             1; // "sky or slime, no lightmap or 256 subdivision"
00033         inline constexpr auto k_MaxLightMaps = 4;
00034         inline constexpr auto k_NumAmbients = 4;
00035
00036         enum Contents
00037         {
00038             Contents_Empty = -1,
00039             Contents_Solid = -2,
00040             Contents_Water = -3,
00041             Contents_Slime = -4,
00042             Contents_Lava = -5,
00043             Contents_Sky = -6,
00044             Contents_Origin = -7, // CSG brush (removed during build)
00045             Contents_Clip = -8,       // Legacy (changed to Solid)
00046             Contents_Current0 = -9,
00047             Contents_Current90 = -10,
00048             Contents_Current180 = -11,
00049             Contents_Current270 = -12,
00050             Contents_CurrentUp = -13,
00051             Contents_CurrentDown = -14,
00052             Contents_Translucent = -15,
00053         };
00054
00055         struct Lump
00056         {
00057             oxyU32 m_fileOffset;
00058             oxyU32 m_length;
00059         };
00060         static_assert(sizeof(Lump) == 8, "Lump struct size is not 8 bytes");
00061         static_assert(alignof(Lump) == 4,
00062                       "Lump struct alignment is not 4 bytes");
00063         static_assert(std::is_trivial_v<Lump>,
00064                       "Lump struct is not a trivial type");
00065         static_assert(offsetof(Lump, m_fileOffset) == 0,
00066                       "Lump struct m_fileOffset offset is not 0");
00067         static_assert(offsetof(Lump, m_length) == 4,
00068                       "Lump struct m_length offset is not 4");
00069
00070         enum LumpIndex
00071         {
00072             LumpIndex_Entities = 0,
00073             LumpIndex_Planes,
00074             LumpIndex_Textures,
00075             LumpIndex_Vertexes,
00076             LumpIndex_Visibility,
00077             LumpIndex_Nodes,
00078             LumpIndex_TexInfo,
00079             LumpIndex_Faces,
00080             LumpIndex_Lighting,
00081             LumpIndex_ClipNodes,
00082             LumpIndex_Leafs,
```

```
00083             LumpIndex_MarkSurfaces,
00084             LumpIndex_Edges,
00085             LumpIndex_SurfEdges,
00086             LumpIndex_Models,
00087             LumpIndex_Count
00088         };
00089
00090         struct Header
00091         {
00092             oxyS32 m_version;
00093             Lump m_lumps[LumpIndex_Count];
00094         };
00095         static_assert(sizeof(Header) == 124,
00096                       "Header struct size is not 124 bytes");
00097         static_assert(alignof(Header) == 4,
00098                       "Header struct alignment is not 4 bytes");
00099         static_assert(std::is_trivial_v<Header>,
00100                       "Header struct is not a trivial type");
00101         static_assert(offsetof(Header, m_version) == 0,
00102                       "Header struct m_version offset is not 0");
00103         static_assert(offsetof(Header, m_lumps) == 4,
00104                       "Header struct m_lumps offset is not 4");
00105
00106         enum PlaneType
00107         {
00108             Plane_X = 0,
00109             Plane_Y,
00110             Plane_Z,
00111             Plane_AnyX,
00112             Plane_AnyY,
00113             Plane_AnyZ
00114         };
00115
00116         struct Plane
00117         {
00118             oxyF32 m_normal[3];
00119             oxyF32 m_dist;
00120             oxyU32 m_type;
00121         };
00122         static_assert(sizeof(Plane) == 20, "Plane struct size is not 20 bytes");
00123         static_assert(alignof(Plane) == 4,
00124                       "Plane struct alignment is not 4 bytes");
00125         static_assert(std::is_trivial_v<Plane>,
00126                       "Plane struct is not a trivial type");
00127         static_assert(offsetof(Plane, m_normal) == 0,
00128                       "Plane struct m_normal offset is not 0");
00129         static_assert(offsetof(Plane, m_dist) == 12,
00130                       "Plane struct m_dist offset is not 12");
00131         static_assert(offsetof(Plane, m_type) == 16,
00132                       "Plane struct m_type offset is not 16");
00133
00134         struct MipTexLump
00135         {
00136             oxyU32 m_numMipTex;
00137             oxyU32 m_dataOffsets[4]; // [m_numMipTex]
00138         };
00139         static_assert(sizeof(MipTexLump) == 20,
00140                       "MipTexLump struct size is not 20 bytes");
00141         static_assert(alignof(MipTexLump) == 4,
00142                       "MipTexLump struct alignment is not 4 bytes");
00143         static_assert(std::is_trivial_v<MipTexLump>,
00144                       "MipTexLump struct is not a trivial type");
00145         static_assert(offsetof(MipTexLump, m_numMipTex) == 0,
00146                       "MipTexLump struct m_numMipTex offset is not 0");
00147         static_assert(offsetof(MipTexLump, m_dataOffsets) == 4,
00148                       "MipTexLump struct m_dataOffsets offset is not 4");
00149
00150         struct MipTex
00151         {
00152             oxyChar m_name[16];
00153             oxyU32 m_width;
00154             oxyU32 m_height;
00155             oxyU32 m_offsets[k_NumMipLevels];
00156         };
00157         static_assert(sizeof(MipTex) == 40,
00158                       "MipTex struct size is not 40 bytes");
00159         static_assert(alignof(MipTex) == 4,
00160                       "MipTex struct alignment is not 4 bytes");
00161         static_assert(std::is_trivial_v<MipTex>,
00162                       "MipTex struct is not a trivial type");
00163         static_assert(offsetof(MipTex, m_name) == 0,
00164                       "MipTex struct m_name offset is not 0");
00165         static_assert(offsetof(MipTex, m_width) == 16,
00166                       "MipTex struct m_width offset is not 16");
00167         static_assert(offsetof(MipTex, m_height) == 20,
00168                       "MipTex struct m_height offset is not 20");
00169         static_assert(offsetof(MipTex, m_offsets) == 24,
```

```
00170                       "MipTex struct m_offsets offset is not 24");
00171
00172         struct Vertex
00173         {
00174             oxyF32 m_position[3];
00175         };
00176         static_assert(sizeof(Vertex) == 12,
00177                       "Vertex struct size is not 12 bytes");
00178         static_assert(alignof(Vertex) == 4,
00179                       "Vertex struct alignment is not 4 bytes");
00180         static_assert(std::is_trivial_v<Vertex>,
00181                       "Vertex struct is not a trivial type");
00182         static_assert(offsetof(Vertex, m_position) == 0,
00183                       "Vertex struct m_position offset is not 0");
00184
00185         struct Node
00186         {
00187             oxyU32 m_planeIndex;
00188             oxyS16 m_children[2]; // Negative numbers are -(leafs+1)
00189             oxyS16 m_mins[3];
00190             oxyS16 m_maxs[3];
00191             oxyU16 m_firstFaceIndex;
00192             oxyU16 m_faceCount;
00193         };
00194         static_assert(sizeof(Node) == 24, "Node struct size is not 32 bytes");
00195         static_assert(alignof(Node) == 4,
00196                       "Node struct alignment is not 4 bytes");
00197         static_assert(std::is_trivial_v<Node>,
00198                       "Node struct is not a trivial type");
00199         static_assert(offsetof(Node, m_planeIndex) == 0,
00200                       "Node struct m_planeIndex offset is not 0");
00201         static_assert(offsetof(Node, m_children) == 4,
00202                       "Node struct m_children offset is not 4");
00203         static_assert(offsetof(Node, m_mins) == 8,
00204                       "Node struct m_mins offset is not 8");
00205         static_assert(offsetof(Node, m_maxs) == 14,
00206                       "Node struct m_maxs offset is not 14");
00207         static_assert(offsetof(Node, m_firstFaceIndex) == 20,
00208                       "Node struct m_firstFaceIndex offset is not 20");
00209         static_assert(offsetof(Node, m_faceCount) == 22,
00210                       "Node struct m_faceCount offset is not 22");
00211
00212         struct TexInfo
00213         {
00214             oxyF32 m_vecs[2][4]; // [s/t][xyz offset]
00215             oxyS32 m_mipTexIndex;
00216             oxyS32 m_flags;
00217         };
00218         static_assert(sizeof(TexInfo) == 40,
00219                       "TexInfo struct size is not 40 bytes");
00220         static_assert(alignof(TexInfo) == 4,
00221                       "TexInfo struct alignment is not 4 bytes");
00222         static_assert(std::is_trivial_v<TexInfo>,
00223                       "TexInfo struct is not a trivial type");
00224         static_assert(offsetof(TexInfo, m_vecs) == 0,
00225                       "TexInfo struct m_vecs offset is not 0");
00226         static_assert(offsetof(TexInfo, m_mipTexIndex) == 32,
00227                       "TexInfo struct m_mipTexIndex offset is not 32");
00228         static_assert(offsetof(TexInfo, m_flags) == 36,
00229                       "TexInfo struct m_flags offset is not 36");
00230
00231         struct Face
00232         {
00233             oxyU16 m_planeIndex;
00234             oxyU16 m_side;
00235             oxyU32 m_firstEdgeIndex;
00236             oxyU16 m_edgeCount;
00237             oxyU16 m_texInfoIndex;
00238             oxyU8 m_lightStyles[k_MaxLightMaps];
00239             oxyU32 m_lightMapOffset;
00240         };
00241         static_assert(sizeof(Face) == 20, "Face struct size is not 20 bytes");
00242         static_assert(alignof(Face) == 4,
00243                       "Face struct alignment is not 4 bytes");
00244         static_assert(std::is_trivial_v<Face>,
00245                       "Face struct is not a trivial type");
00246         static_assert(offsetof(Face, m_planeIndex) == 0,
00247                       "Face struct m_planeIndex offset is not 0");
00248         static_assert(offsetof(Face, m_side) == 2,
00249                       "Face struct m_side offset is not 2");
00250         static_assert(offsetof(Face, m_firstEdgeIndex) == 4,
00251                       "Face struct m_firstEdgeIndex offset is not 4");
00252         static_assert(offsetof(Face, m_edgeCount) == 8,
00253                       "Face struct m_edgeCount offset is not 8");
00254         static_assert(offsetof(Face, m_texInfoIndex) == 10,
00255                       "Face struct m_texInfoIndex offset is not 10");
00256         static_assert(offsetof(Face, m_lightStyles) == 12,
```

```
00257                   "Face struct m_lightStyles offset is not 12");
00258         static_assert(offsetof(Face, m_lightMapOffset) == 16,
00259                   "Face struct m_lightMapOffset offset is not 16");
00260
00261         struct ClipNode
00262         {
00263             oxyU32 m_planeIndex;
00264             oxyS16 m_children[2]; // Negatives are contents
00265         };
00266         static_assert(sizeof(ClipNode) == 8,
00267                   "ClipNode struct size is not 8 bytes");
00268         static_assert(alignof(ClipNode) == 4,
00269                   "ClipNode struct alignment is not 4 bytes");
00270         static_assert(std::is_trivial_v<ClipNode>,
00271                   "ClipNode struct is not a trivial type");
00272         static_assert(offsetof(ClipNode, m_planeIndex) == 0,
00273                   "ClipNode struct m_planeIndex offset is not 0");
00274         static_assert(offsetof(ClipNode, m_children) == 4,
00275                   "ClipNode struct m_children offset is not 4");
00276
00277         // "leaf 0 is the generic CONTENTS_SOLID leaf"
00278         struct Leaf
00279         {
00280             oxyS32 m_contents;
00281             oxyS32 m_visOffset; // -1 = none
00282             oxyS16 m_mins[3];   // "for frustum culling"
00283             oxyS16 m_maxs[3];
00284             oxyU16 m_firstMarkSurfaceIndex;
00285             oxyU16 m_markSurfaceCount;
00286             oxyU8 m_ambientLevels[k_NumAmbients];
00287         };
00288         static_assert(sizeof(Leaf) == 28, "Leaf struct size is not 28 bytes");
00289         static_assert(alignof(Leaf) == 4,
00290                   "Leaf struct alignment is not 4 bytes");
00291         static_assert(std::is_trivial_v<Leaf>,
00292                   "Leaf struct is not a trivial type");
00293         static_assert(offsetof(Leaf, m_contents) == 0,
00294                   "Leaf struct m_contents offset is not 0");
00295         static_assert(offsetof(Leaf, m_visOffset) == 4,
00296                   "Leaf struct m_visOffset offset is not 4");
00297         static_assert(offsetof(Leaf, m_mins) == 8,
00298                   "Leaf struct m_mins offset is not 8");
00299         static_assert(offsetof(Leaf, m_maxs) == 14,
00300                   "Leaf struct m_maxs offset is not 14");
00301         static_assert(offsetof(Leaf, m_firstMarkSurfaceIndex) == 20,
00302                   "Leaf struct m_firstMarkSurfaceIndex offset is not 20");
00303         static_assert(offsetof(Leaf, m_markSurfaceCount) == 22,
00304                   "Leaf struct m_markSurfaceCount offset is not 22");
00305         static_assert(offsetof(Leaf, m_ambientLevels) == 24,
00306                   "Leaf struct m_ambientLevels offset is not 24");
00307
00308         struct Edge
00309         {
00310             oxyU16 m_vertexIndices[2];
00311         };
00312         static_assert(sizeof(Edge) == 4, "Edge struct size is not 4 bytes");
00313         static_assert(alignof(Edge) == 2,
00314                   "Edge struct alignment is not 2 bytes");
00315         static_assert(std::is_trivial_v<Edge>,
00316                   "Edge struct is not a trivial type");
00317         static_assert(offsetof(Edge, m_vertexIndices) == 0,
00318                   "Edge struct m_vertexIndices offset is not 0");
00319
00320         struct Model
00321         {
00322             oxyF32 m_mins[3];
00323             oxyF32 m_maxs[3];
00324             oxyF32 m_origin[3];
00325             oxyU32 m_headNodes[k_MaxMapHulls];
00326             oxyU32 m_visLeafs; // "not including the solid leaf 0"
00327             oxyU32 m_firstFaceIndex;
00328             oxyU32 m_faceCount;
00329         };
00330         static_assert(sizeof(Model) == 64, "Model struct size is not 64 bytes");
00331         static_assert(alignof(Model) == 4,
00332                   "Model struct alignment is not 4 bytes");
00333         static_assert(std::is_trivial_v<Model>,
00334                   "Model struct is not a trivial type");
00335         static_assert(offsetof(Model, m_mins) == 0,
00336                   "Model struct m_mins offset is not 0");
00337         static_assert(offsetof(Model, m_maxs) == 12,
00338                   "Model struct m_maxs offset is not 12");
00339         static_assert(offsetof(Model, m_origin) == 24,
00340                   "Model struct m_origin offset is not 24");
00341         static_assert(offsetof(Model, m_headNodes) == 36,
00342                   "Model struct m_headNodes offset is not 36");
00343         static_assert(offsetof(Model, m_visLeafs) == 52,
```

```
00344                       "Model struct m_visLeafs offset is not 52");
00345         static_assert(offsetof(Model, m_firstFaceIndex) == 56,
00346                       "Model struct m_firstFaceIndex offset is not 56");
00347         static_assert(offsetof(Model, m_faceCount) == 60,
00348                       "Model struct m_faceCount offset is not 60");
00349
00350     }; // namespace BSPDefines
00351
00352     struct BSPWorldData : NonCopyable
00353     {
00354         std::vector<std::unordered_map<std::string, std::string» m_entitiesText;
00355         std::vector<BSPDefines::Plane> m_planes;
00356         std::vector<BSPDefines::MipTex> m_miptex;
00357         std::vector<BSPDefines::Vertex> m_vertices;
00358         std::vector<oxyU8> m_visibility;
00359         std::vector<BSPDefines::Node> m_nodes;
00360         std::vector<BSPDefines::TexInfo> m_texinfo;
00361         std::vector<BSPDefines::Face> m_faces;
00362         std::vector<BSPDefines::ClipNode> m_clipNodes;
00363         std::vector<BSPDefines::Leaf> m_leaves;
00364         std::vector<oxyU16> m_marksurfaces;
00365         std::vector<BSPDefines::Edge> m_edges;
00366         std::vector<oxyS32> m_surfedges;
00367         std::vector<BSPDefines::Model> m_models;
00368
00369         auto Load(std::string_view mapname) -> oxyBool;
00370     };
00371
00372 }; // namespace oxygen
```

7.95 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/↩ World.cc File Reference

```
#include "OxygenPCH.h"
#include "World.h"
#include "Entity/Entity.h"
#include "Component/HullComponent/HullComponent.h"
#include "Component/CameraComponent/CameraComponent.h"
#include "Component/EnvPushComponent/EnvPushComponent.h"
#include "Gfx/GfxRenderer.h"
#include "Input/InputManager.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

7.96 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/↩ World.h File Reference

```
#include "BSP.h"
```

Classes

- struct oxygen::World
- struct oxygen::World::LineTraceResult

Namespaces

- namespace oxygen

7.97 World.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "BSP.h"
00004
00005 namespace oxygen
00006 {
00007     struct World final : ManagedObject
00008     {
00009         OXYGENOBJECT(World, ManagedObject);
00010
00011         std::unique_ptr<const BSPWorldData> m_bspData{};
00012
00013         auto
00014         GetEntityList() const -> const std::vector<std::shared_ptr<struct Entity»&
00015         {
00016             return m_entities;
00017         }
00018
00019         auto RandomPlayerSpawn() const -> oxyVec3;
00020
00021         auto FindLeaf(const oxyVec3& position,
00022                       oxySize modelIndex) const -> const BSPDefines::Leaf*;
00023
00024         struct LineTraceResult
00025         {
00026             oxyBool m_allSolid{};
00027             oxyBool m_startSolid{};
00028             oxyF32 m_fraction{};
00029             oxyVec3 m_endPos{};
00030             oxyVec3 m_planeNormal{};
00031             oxyF32 m_planeDist{};
00032             std::shared_ptr<struct Entity> m_hitEntity{};
00033         };
00034         auto HullTrace(CollisionHull hull, const oxyVec3& start,
00035                        const oxyVec3& end,
00036                        LineTraceResult& result) const -> oxyBool;
00037         auto LineTrace(const oxyVec3& start, const oxyVec3& end,
00038                        const struct Entity* self,
00039                        LineTraceResult& result) const -> oxyBool;
00040
00041         auto CalculateHullSlideMovement(CollisionHull hull,
00042                                         const oxyVec3& position,
00043                                         const oxyVec3& distance) -> oxyVec3;
00044
00045         auto SpawnEntity(oxyObjectID id = 0) -> std::shared_ptr<Entity>;
00046
00047         auto RemoveEntity(struct Entity* ent) -> void;
00048
00049         auto GetLocalPlayer() const -> std::weak_ptr<Entity>
00050         {
00051             return m_localPlayer;
00052         }
00053         auto SetLocalPlayer(std::shared_ptr<Entity> player) -> void;
00054
00055       private:
00056         friend struct GameManager;
00057         friend auto LoadWorld(std::string_view name) -> std::shared_ptr<World>;
00058         auto SubmitBSPFacesToRenderQueue() -> void;
00059
00060         oxyVec3 m_renderCameraPosition{};
00061
00062         std::weak_ptr<struct Entity> m_localPlayer{};
00063         std::vector<std::shared_ptr<struct Entity» m_entities;
00064
00065         std::vector<std::shared_ptr<const struct GfxTexture» m_bspTextures;
00066         std::shared_ptr<const struct GfxTexture> m_lightmapTexture;
00067         std::vector<std::array<oxyU32, 4» m_lightmapRects;
00068         oxyU32 m_lightmapSampleSize{};
00069         oxyU32 m_lightmapBlockWidth{};
00070         oxyU32 m_lightmapBlockHeight{};
00071         oxyU32 m_lightmapNumRects{};
00072         struct WorldTri
00073         {
```

```
00074             oxyVec3 m_vertices[3];
00075             oxyVec2 m_texcoords[3];
00076             oxyVec2 m_lmtexcoords[3];
00077             oxyU32 m_textureIndex{};
00078         };
00079         std::vector<std::vector<WorldTri» m_bspFaces;
00080         std::vector<oxyVec3> m_playerStarts;
00081         std::vector<oxyU8> m_cameraPVS;
00082         std::vector<oxyS16> m_bspNodeParents;
00083         std::vector<oxyS16> m_bspLeafParents;
00084         std::bitset<BSPDefines::k_MaxMapNodes> m_nodesMarkedForRender;
00085         std::bitset<BSPDefines::k_MaxMapFaces> m_facesMarkedForRender;
00086 
00087         //auto SummonPlayer(const EntitySummonParams& params)
00088         //  -> std::shared_ptr<Entity>;
00089         //auto SummonGrenadeProjectile(const EntitySummonParams& params)
00090         //  -> std::shared_ptr<Entity>;
00091         //auto SummonDebugCube(const EntitySummonParams& params)
00092         //  -> std::shared_ptr<Entity>;
00093 
00094         auto CreateEntitiesFromBSP() -> void;
00095 
00096         auto
00097         TestBoundsIntersectVisibleNodes(const oxyVec3& mins,
00098                                         const oxyVec3& maxs) const -> oxyBool;
00099 
00100         auto RenderTraverseBSPNode(oxyS32 nodeIndex,
00101                                    const oxyVec3& origin) -> void;
00102         auto RenderBSPLeaf(const BSPDefines::Leaf& leaf,
00103                            const oxyVec3& origin) -> void;
00104         auto RenderBSPFace(struct GfxRenderer& gfx, oxySize faceindex,
00105                            const oxyVec3& origin) -> void;
00106 
00107         auto ComputeTriFaces() -> void;
00108 
00109         auto MarkPVSNodesFromLeaf(const BSPDefines::Leaf* leaf,
00110                                   oxySize modelIndex) -> void;
00111 
00112         auto RecursiveClipNodeLineTrace(oxyS32 clipNodeIndex,
00113                                         const oxyVec3& start,
00114                                         const oxyVec3& end,
00115                                         LineTraceResult& result) const -> bool;
00116         auto RecursiveNodeLineTrace(oxyS32 nodeIndex, const oxyVec3& start,
00117                                     const oxyVec3& end,
00118                                     LineTraceResult& result) const -> bool;
00119         auto RecursiveSlideHull(oxyS32 rootClipNode, const oxyVec3& position,
00120                                 const oxyVec3& offset, int depth) -> oxyVec3;
00121         auto Update(float deltaTimeSeconds) -> void;
00122     };
00123 }; // namespace oxygen
```

7.98 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/WorldLoader.cc File Reference

```
#include "OxygenPCH.h"
#include "WorldLoader.h"
#include "World.h"
#include "BSP.h"
#include "Gfx/GfxRenderer.h"
#include "Platform/Platform.h"
```

Namespaces

- namespace oxygen

Functions

- auto oxygen::LoadWorld (std::string_view name) -> std::shared_ptr< World >

7.99 C:/Users/a/Desktop/arihawe-oxygen2/oxygen/codebase/World/↩ WorldLoader.h File Reference

Namespaces

- namespace oxygen

Functions

- auto oxygen::LoadWorld (std::string_view name) -> std::shared_ptr< World >

7.100 WorldLoader.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 namespace oxygen
00004 {
00005     auto LoadWorld(std::string_view name) -> std::shared_ptr<struct World>;
00006 }; // namespace oxygen
```

Index