MAGE

Generated by Doxygen 1.8.12

Contents

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

Namespace Documentation

4.1 mage Namespace Reference

Classes

- struct AABB
- struct BS
- class Camera
- class CameraNode
- struct CartesianAxesSystem
- struct CartesianCoordinateSystem
- class ConditionVariable
- struct DDS_HEADER
- struct DDS_HEADER_DXT10
- struct DDS_PIXELFORMAT
- class DeviceEnumeration
- struct Edge
- class Engine
- struct EngineSetup
- struct Face
- class Font
- struct IndexedEdge
- struct IndexedFace
- class InputManager
- class Keyboard
- class Loadable
- struct LoggingConfiguration
- struct LVertex
- class MainWindow
- class MemoryArena
- class Mouse
- class Mutex
- struct MutexLock
- class OrthographicCamera
- class ParallelForLoop
- class PerspectiveCamera
- class ProgressReporter
- class ReadWriteMutex

- struct ReadWriteMutexLock
- class Reference
- class ReferenceCounted
- class Renderer
- class Resource
- class ResourceManager
- class SceneNode
- class SceneNodeVisitor
- class Semaphore
- class State
- class StateManager
- class Task
- class Timer
- struct TLVertex
- struct Transform
- struct Variable
- class VariableScript
- struct Vertex
- struct ViewerSetup

Typedefs

- typedef XMFLOAT3 float3
- typedef XMFLOAT4 float4
- typedef XMFLOAT4 colour

Enumerations

- enum ReadWriteMutexLockType { READ, WRITE }
- enum VariableType {
  BoolType, IntType, FloatType, Float3Type,
  Float4Type, ColourType, StringType, UnknownType }
- enum DDS_MISC_FLAGS2 { DDS_MISC_FLAGS2_ALPHA_MODE_MASK = 0x7L }
- enum DDS_ALPHA_MODE {
  DDS_ALPHA_MODE_UNKNOWN = 0, DDS_ALPHA_MODE_STRAIGHT = 1, DDS_ALPHA_MODE_PR←
  EMULTIPLIED = 2, DDS_ALPHA_MODE_OPAQUE = 3,
  DDS_ALPHA_MODE_CUSTOM = 4 }

Functions

- void PrintConsoleHeader ()
- static const char ∗ FindWordEnd (const char ∗buffer)
- static void ProcessError (const char ∗format, const va_list args, const string &error_type, int error_disposition)
- void Debug (const char ∗format,...)
- void Info (const char ∗format,...)
- void Warning (const char ∗format,...)
- void Error (const char ∗format,...)
- void Fatal (const char ∗format,...)
- HRESULT InitializeConsole ()
- int TerminalWidth ()
- AABB Union (const AABB &aabb, const XMFLOAT3 &point)
- AABB Union (const AABB &aabb1, const AABB &aabb2)

- AABB Overlap (const AABB &aabb1, const AABB &aabb2)
- AABB OverlapStrict (const AABB &aabb1, const AABB &aabb2)
- void * AllocAligned (size_t size)
- void FreeAligned (void *ptr)
- template< typename T >
  T * AtomicCompareAndSwapPointer (T **destination, T *exchange, T *comparand)
- int32_t AtomicAdd (AtomicInt32 *addend, int32_t value)
- int32_t AtomicCompareAndSwap (AtomicInt32 *destination, int32_t exchange, int32_t comparand)
- float AtomicAdd (volatile float *addend, float value)
- size_t NumberOfSystemCores ()
- void ParallelFor (function< void(size_t) > func, size_t nb_work, size_t chunk_size=1)
- size_t MaxThreadIndex ()
- void ParallelInit ()
- void ParallelCleanup ()
- static DWORD WINAPI task_entry (LPVOID lpParameter)
- void TasksInit ()
- void TasksCleanup ()
- void EnqueueTasks (const vector< Task * > &tasks)
- void WaitForAllTasks ()
- INT_PTR CALLBACK SettingsDialogProcDelegate (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
- bool RejectDisplayMode (const DXGI_MODE_DESC1 *display_mode_desc)
- size_t BitsPerPixel (DXGI_FORMAT format)
- static HRESULT LoadTextureDataFromFile (_In_z_ const wchar_t *file_name, std::unique_ptr< uint8_t[ ]> &dds_data, DDS_HEADER **header, uint8_t **bit_data, size_t *bit_size)
- static void GetSurfaceInfo (_In_ size_t width, _In_ size_t height, _In_ DXGI_FORMAT fmt, _Out_opt_ size_t *out_nb_bytes, _Out_opt_ size_t *out_row_bytes, _Out_opt_ size_t *out_nb_rows)
- static DXGI_FORMAT GetDXGIFormat (const DDS_PIXELFORMAT &ddpf)
- static DXGI_FORMAT MakeSRGB (_In_ DXGI_FORMAT format)
- static HRESULT FillInitData (_In_ size_t width, _In_ size_t height, _In_ size_t depth, _In_ size_t mip_count, _In_ size_t array_size, _In_ DXGI_FORMAT format, _In_ size_t maxsize, _In_ size_t bit_size, _In_reads_bytes_(bit_size) const uint8_t *bit_data, _Out_ size_t &twidth, _Out_ size_t &theight, _Out_ size_t &tdepth, _Out_ size_t &skip_mip, _Out_writes_(mip_count *array_size) D3D11_SUBRESOURCE_DATA *init_data)
- static HRESULT CreateD3DResources (_In_ ID3D11Device *d3dDevice, _In_ uint32_t res_dim, _In_ size_t width, _In_ size_t height, _In_ size_t depth, _In_ size_t mip_count, _In_ size_t array_size, _In_ DXGI_FORMAT format, _In_ D3D11_USAGE usage, _In_ uint32_t bindFlags, _In_ uint32_t cpu_access_flags, _In_ uint32_t misc_flags, _In_ bool forceSRGB, _In_ bool is_cube_map, _In_reads_opt_(mip_count *array_size) D3D11_SUBRESOURCE_DATA *init_data, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view)
- static HRESULT CreateTextureFromDDS (_In_ ID3D11Device *d3dDevice, _In_opt_ ID3D11DeviceContext *d3dContext, _In_ const DDS_HEADER *header, _In_reads_bytes_(bit_size) const uint8_t *bit_data, _In_ size_t bit_size, _In_ size_t maxsize, _In_ D3D11_USAGE usage, _In_ uint32_t bindFlags, _In_ uint32_t cpu_access_flags, _In_ uint32_t misc_flags, _In_ bool forceSRGB, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view)
- static DDS_ALPHA_MODE GetAlphaMode (_In_ const DDS_HEADER *header)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromMemory (ID3D11Device *d3dDevice, const uint8_t *dds_data, size_t dds_dataSize, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, size_t maxsize, DDS_ALPHA_MODE *alpha_mode)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromMemory (ID3D11Device *d3dDevice, ID3D11DeviceContext *d3dContext, const uint8_t *dds_data, size_t dds_dataSize, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, size_t maxsize, DDS_ALPHA_MODE *alpha_mode)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromMemoryEx (ID3D11Device *d3dDevice, const uint8_t *dds_data, size_t dds_dataSize, size_t maxsize, D3D11_USAGE usage, uint32_t bindFlags, uint32_t cpu_access_flags, uint32_t misc_flags, bool forceSRGB, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, DDS_ALPHA_MODE *alpha_mode)

- _Use_decl_annotations_ HRESULT CreateDDSTextureFromMemoryEx (ID3D11Device *d3dDevice, ID3D11DeviceContext *d3dContext, const uint8_t *dds_data, size_t dds_dataSize, size_t maxsize, D3D11_USAGE usage, uint32_t bindFlags, uint32_t cpu_access_flags, uint32_t misc_flags, bool forceSRGB, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, DDS_ALPHA_MODE *alpha_mode)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromFile (ID3D11Device *d3dDevice, const wchar_t *file_name, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, size_t maxsize, DDS_ALPHA_MODE *alpha_mode)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromFile (ID3D11Device *d3dDevice, ID3D11DeviceContext *d3dContext, const wchar_t *file_name, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, size_t maxsize, DDS_ALPHA_MODE *alpha_mode)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromFileEx (ID3D11Device *d3dDevice, const wchar_t *file_name, size_t maxsize, D3D11_USAGE usage, uint32_t bindFlags, uint32_t cpu_access_flags, uint32_t misc_flags, bool forceSRGB, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, DDS_ALPHA_MODE *alpha_mode)
- _Use_decl_annotations_ HRESULT CreateDDSTextureFromFileEx (ID3D11Device *d3dDevice, ID3D11DeviceContext *d3dContext, const wchar_t *file_name, size_t maxsize, D3D11_USAGE usage, uint32_t bindFlags, uint32_t cpu_access_flags, uint32_t misc_flags, bool forceSRGB, ID3D11Resource **texture, ID3D11ShaderResourceView **texture_view, DDS_ALPHA_MODE *alpha_mode)
- HRESULT CreateDDSTextureFromMemory (_In_ ID3D11Device *d3dDevice, _In_reads_bytes_(dds_dataSize) const uint8_t *dds_data, _In_ size_t dds_dataSize, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _In_ size_t maxsize=0, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromFile (_In_ ID3D11Device *d3dDevice, _In_z_ const wchar_t *szFileName, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _In_ size_t maxsize=0, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromMemory (_In_ ID3D11Device *d3dDevice, _In_opt_ ID3D11DeviceContext *d3dContext, _In_reads_bytes_(dds_dataSize) const uint8_t *dds_data, _In_ size_t dds_dataSize, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _In_ size_t maxsize=0, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromFile (_In_ ID3D11Device *d3dDevice, _In_opt_ ID3D11DeviceContext *d3dContext, _In_z_ const wchar_t *szFileName, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _In_ size_t maxsize=0, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromMemoryEx (_In_ ID3D11Device *d3dDevice, _In_reads_bytes_(dds_dataSize) const uint8_t *dds_data, _In_ size_t dds_dataSize, _In_ size_t maxsize, _In_ D3D11_USAGE usage, _In_ uint32_t bindFlags, _In_ uint32_t cpu_access_flags, _In_ uint32_t misc_flags, _In_ bool forceSRGB, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromFileEx (_In_ ID3D11Device *d3dDevice, _In_z_ const wchar_t *szFileName, _In_ size_t maxsize, _In_ D3D11_USAGE usage, _In_ uint32_t bindFlags, _In_ uint32_t cpu_access_flags, _In_ uint32_t misc_flags, _In_ bool forceSRGB, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromMemoryEx (_In_ ID3D11Device *d3dDevice, _In_opt_ ID3D11DeviceContext *d3dContext, _In_reads_bytes_(dds_dataSize) const uint8_t *dds_data, _In_ size_t dds_dataSize, _In_ size_t maxsize, _In_ D3D11_USAGE usage, _In_ uint32_t bindFlags, _In_ uint32_t cpu_access_flags, _In_ uint32_t misc_flags, _In_ bool forceSRGB, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- HRESULT CreateDDSTextureFromFileEx (_In_ ID3D11Device *d3dDevice, _In_opt_ ID3D11DeviceContext *d3dContext, _In_z_ const wchar_t *szFileName, _In_ size_t maxsize, _In_ D3D11_USAGE usage, _In_ uint32_t bindFlags, _In_ uint32_t cpu_access_flags, _In_ uint32_t misc_flags, _In_ bool forceSRGB, _Outptr_opt_ ID3D11Resource **texture, _Outptr_opt_ ID3D11ShaderResourceView **texture_view, _Out_opt_ DDS_ALPHA_MODE *alpha_mode=nullptr)
- void ComboBoxAdd (HWND dialog, int id, const void *data, const wchar_t *desc)
- void ComboBoxSelect (HWND dialog, int id, int index)
- void ComboBoxSelect (HWND dialog, int id, const void *data)
- const void * ComboBoxSelected (HWND dialog, int id)

- bool ComboBoxSomethingSelected (HWND dialog, int id)
- int ComboBoxCount (HWND dialog, int id)
- bool ComboBoxContains (HWND dialog, int id, const wchar_t ∗desc)
- LRESULT CALLBACK MainWindowProc (HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam)

Variables

- LoggingConfiguration g_logging_configuration
- Engine ∗ g_engine = nullptr
- const D3D11_INPUT_ELEMENT_DESC vertex_input_element_desc [ ]
- const D3D11_INPUT_ELEMENT_DESC lvertex_input_element_desc [ ]
- const D3D11_INPUT_ELEMENT_DESC tlvertex_input_element_desc [ ]
- static vector< thread > workers
- static bool shutdown_workers = false
- static ParallelForLoop ∗ work_list = nullptr
- static mutex work_list_mutex
- static condition_variable work_list_condition
- thread_local size_t ThreadIndex
- static HANDLE ∗ threads
- static Mutex ∗ task_queue_mutex = Mutex::Create()
- static vector< Task ∗> task_queue
- static Semaphore ∗ worker_semaphore
- static uint32_t nb_unfinished_tasks
- static ConditionVariable ∗ tasks_running_condition
- DeviceEnumeration ∗ g_device_enumeration = nullptr
- const D3D_FEATURE_LEVEL g_feature_levels [ ]
- const DXGI_FORMAT g_pixel_formats [ ]
- const uint32_t DDS_MAGIC = 0x20534444

4.1.1 Typedef Documentation

4.1.1.1 colour

```
typedef XMFLOAT4 mage::colour
```

4.1.1.2 float3

```
typedef XMFLOAT3 mage::float3
```

4.1.1.3 float4

```
typedef XMFLOAT4 mage::float4
```

4.1.2 Enumeration Type Documentation

4.1.2.1 DDS_ALPHA_MODE

```
enum mage::DDS_ALPHA_MODE
```

Enumerator

DDS_ALPHA_MODE_UNKNOWN	
DDS_ALPHA_MODE_STRAIGHT	
DDS_ALPHA_MODE_PREMULTIPLIED	
DDS_ALPHA_MODE_OPAQUE	
DDS_ALPHA_MODE_CUSTOM	

4.1.2.2 DDS_MISC_FLAGS2

```
enum mage::DDS_MISC_FLAGS2
```

Enumerator

DDS_MISC_FLAGS2_ALPHA_MODE_MASK	

4.1.2.3 ReadWriteMutexLockType

```
enum mage::ReadWriteMutexLockType
```

Type of read write mutex locks.

Enumerator

READ	
WRITE	

4.1.2.4 VariableType

```
enum mage::VariableType
```

Enumeration of variable types.

Enumerator

BoolType	
IntType	
FloatType	
Float3Type	
Float4Type	
ColourType	
StringType	
UnknownType	

4.1.3 Function Documentation

4.1.3.1 AllocAligned()

```
T * mage::AllocAligned (
            size_t count )
```

Allocates memory on an alignment boundary of 64 bytes of the given size.

Parameters

in	*size*	The requested size in bytes to allocate in memory.

Returns

`nullptr` if the allocation failed.
A pointer to the memory block that was allocated. The pointer is a multiple of the alignment of 64 bytes.

Allocates memory on an alignment boundary of 64 bytes.

Template Parameters

T	The type of objects to allocate in memory.

Parameters

in	*count*	The number of objects of type `T` to allocate in memory.

Returns

`nullptr` if the allocation failed.
A pointer to the memory block that was allocated. The pointer is a multiple of the alignment of 64 bytes.

4.1.3.2 AtomicAdd() [1/2]

```
int32_t mage::AtomicAdd (
            AtomicInt32 * addend,
            int32_t value )
```

Performs an atomic addition operation on the specified values.

Parameters

in,out	*addend*	A pointer to the first operand. This value will be replaced with the result of the operation.
in	*value*	The second operand.

Returns

The function returns the result of the operation.

4.1.3.3 AtomicAdd() [2/2]

```
float mage::AtomicAdd (
            volatile float * addend,
            float value )
```

Performs an atomic addition operation on the specified values.

Parameters

in,out	*addend*	A pointer to the first operand. This value will be replaced with the result of the operation.
in	*value*	The second operand.

Returns

The function returns the result of the operation.

4.1.3.4 AtomicCompareAndSwap()

```
int32_t mage::AtomicCompareAndSwap (
            AtomicInt32 * destination,
            int32_t exchange,
            int32_t comparand )
```

Performs an atomic compare-and-exchange operation on the specified values. The function compares the original value against a given comparand value and exchanges the original value with a given exchange value in case of equality.

Parameters

in,out	*destination*	
in	*exchange*	The exchange value.
in	*comparand*	The value to compare to *destination*.

Returns

The function returns the initial value of *destination*.

4.1.3.5 AtomicCompareAndSwapPointer()

```
template<typename T >
T* mage::AtomicCompareAndSwapPointer (
            T ** destination,
            T * exchange,
            T * comparand )
```

Performs an atomic compare-and-exchange operation on the specified pointers. The function compares the original pointer against a given comparand pointer and exchanges the original pointer with a given exchange pointer in case of equality.

Parameters

in,out	*destination*	
in	*exchange*	The exchange pointer.
in	*comparand*	The pointer to compare to *destination*.

Returns

The function returns the initial pointer of *destination*.

4.1.3.6 BitsPerPixel()

```
size_t mage::BitsPerPixel (
            DXGI_FORMAT format )
```

Returns the number of bits per pixel of the given format.

Returns

The number of bits per pixel of the given format.

4.1.3.7 ComboBoxAdd()

```
void mage::ComboBoxAdd (
            HWND dialog,
            int id,
            const void * data,
            const wchar_t * desc )
```

Adds an item associated with the given data and described with the given descriptor to a combo box.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.
in	*data*	A pointer to the data of the item to add.
in	*desc*	The description of the item to add.

4.1.3.8 ComboBoxContains()

```
bool mage::ComboBoxContains (
            HWND dialog,
            int id,
            const wchar_t * desc )
```

Checks whether a combo box contains the given descriptor.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.
in	*desc*	The string description to check.

Returns

`true` if the given description is contained in the combo box. `false` otherwise.

4.1.3.9 ComboBoxCount()

```
int mage::ComboBoxCount (
            HWND dialog,
            int id )
```

Returns the number of items in a combo box.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.

Returns

The number of items of a combo box.

4.1.3.10 ComboBoxSelect() [1/2]

```
void mage::ComboBoxSelect (
            HWND dialog,
            int id,
            int index )
```

Selects the item at the given index in a combo box.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.
in	*index*	The index of the item.

4.1.3.11 ComboBoxSelect() [2/2]

```
void mage::ComboBoxSelect (
            HWND dialog,
            int id,
            const void * data )
```

Selects the item associated with the given data in a combo box.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.
in	*data*	A pointer to the data of the item.

4.1.3.12 ComboBoxSelected()

```
const void * mage::ComboBoxSelected (
            HWND dialog,
            int id )
```

Returns the data associated with the selected item in a combo box.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.

Returns

`nullptr` if the combo box has no items.
A pointer to the data associated with the selected item in the combo box.

4.1.3.13 ComboBoxSomethingSelected()

```
bool mage::ComboBoxSomethingSelected (
            HWND dialog,
            int id )
```

Checks whether a valid item is selected in a combo box.

Parameters

in	*dialog*	A handle to the dialog box that contains the control.
in	*id*	The identifier of the control to be retrieved.

Returns

`true` if a valid item is selected in the combo box. `false` otherwise.

4.1.3.14 CreateD3DResources()

```
static HRESULT mage::CreateD3DResources (
            _In_ ID3D11Device * d3dDevice,
            _In_ uint32_t res_dim,
            _In_ size_t width,
            _In_ size_t height,
            _In_ size_t depth,
            _In_ size_t mip_count,
            _In_ size_t array_size,
            _In_ DXGI_FORMAT format,
            _In_ D3D11_USAGE usage,
            _In_ uint32_t bindFlags,
            _In_ uint32_t cpu_access_flags,
            _In_ uint32_t misc_flags,
            _In_ bool forceSRGB,
            _In_ bool is_cube_map,
            _In_reads_opt_(mip_count *array_size) D3D11_SUBRESOURCE_DATA * init_data,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view )  [static]
```

4.1.3.15 CreateDDSTextureFromFile() [1/4]

```
HRESULT mage::CreateDDSTextureFromFile (
            _In_ ID3D11Device * d3dDevice,
            _In_z_ const wchar_t * szFileName,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _In_ size_t maxsize = 0,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.16 CreateDDSTextureFromFile() [2/4]

```
HRESULT mage::CreateDDSTextureFromFile (
            _In_ ID3D11Device * d3dDevice,
            _In_opt_ ID3D11DeviceContext * d3dContext,
            _In_z_ const wchar_t * szFileName,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _In_ size_t maxsize = 0,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.17 CreateDDSTextureFromFile() [3/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromFile (
            ID3D11Device * d3dDevice,
            const wchar_t * file_name,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            size_t maxsize,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.18 CreateDDSTextureFromFile() [4/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromFile (
            ID3D11Device * d3dDevice,
            ID3D11DeviceContext * d3dContext,
            const wchar_t * file_name,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            size_t maxsize,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.19 CreateDDSTextureFromFileEx() [1/4]

```
HRESULT mage::CreateDDSTextureFromFileEx (
            _In_ ID3D11Device * d3dDevice,
            _In_z_ const wchar_t * szFileName,
            _In_ size_t maxsize,
            _In_ D3D11_USAGE usage,
            _In_ uint32_t bindFlags,
            _In_ uint32_t cpu_access_flags,
            _In_ uint32_t misc_flags,
            _In_ bool forceSRGB,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.20 CreateDDSTextureFromFileEx() [2/4]

```
HRESULT mage::CreateDDSTextureFromFileEx (
            _In_ ID3D11Device * d3dDevice,
            _In_opt_ ID3D11DeviceContext * d3dContext,
            _In_z_ const wchar_t * szFileName,
            _In_ size_t maxsize,
            _In_ D3D11_USAGE usage,
            _In_ uint32_t bindFlags,
            _In_ uint32_t cpu_access_flags,
            _In_ uint32_t misc_flags,
            _In_ bool forceSRGB,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.21 CreateDDSTextureFromFileEx() [3/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromFileEx (
            ID3D11Device * d3dDevice,
            const wchar_t * file_name,
            size_t maxsize,
            D3D11_USAGE usage,
            uint32_t bindFlags,
            uint32_t cpu_access_flags,
            uint32_t misc_flags,
            bool forceSRGB,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.22 CreateDDSTextureFromFileEx() [4/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromFileEx (
            ID3D11Device * d3dDevice,
            ID3D11DeviceContext * d3dContext,
            const wchar_t * file_name,
            size_t maxsize,
            D3D11_USAGE usage,
            uint32_t bindFlags,
            uint32_t cpu_access_flags,
            uint32_t misc_flags,
            bool forceSRGB,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.23 CreateDDSTextureFromMemory() [1/4]

```
HRESULT mage::CreateDDSTextureFromMemory (
            _In_ ID3D11Device * d3dDevice,
            _In_reads_bytes_(dds_dataSize) const uint8_t * dds_data,
            _In_ size_t dds_dataSize,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _In_ size_t maxsize = 0,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.24 CreateDDSTextureFromMemory() [2/4]

```
HRESULT mage::CreateDDSTextureFromMemory (
            _In_ ID3D11Device * d3dDevice,
            _In_opt_ ID3D11DeviceContext * d3dContext,
            _In_reads_bytes_(dds_dataSize) const uint8_t * dds_data,
            _In_ size_t dds_dataSize,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _In_ size_t maxsize = 0,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.25 CreateDDSTextureFromMemory() [3/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromMemory (
            ID3D11Device * d3dDevice,
            const uint8_t * dds_data,
            size_t dds_dataSize,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            size_t maxsize,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.26 CreateDDSTextureFromMemory() [4/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromMemory (
            ID3D11Device * d3dDevice,
            ID3D11DeviceContext * d3dContext,
            const uint8_t * dds_data,
            size_t dds_dataSize,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            size_t maxsize,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.27 CreateDDSTextureFromMemoryEx() [1/4]

```
HRESULT mage::CreateDDSTextureFromMemoryEx (
            _In_ ID3D11Device * d3dDevice,
            _In_reads_bytes_(dds_dataSize) const uint8_t * dds_data,
            _In_ size_t dds_dataSize,
            _In_ size_t maxsize,
            _In_ D3D11_USAGE usage,
            _In_ uint32_t bindFlags,
            _In_ uint32_t cpu_access_flags,
            _In_ uint32_t misc_flags,
            _In_ bool forceSRGB,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.28 CreateDDSTextureFromMemoryEx() [2/4]

```
HRESULT mage::CreateDDSTextureFromMemoryEx (
            _In_ ID3D11Device * d3dDevice,
            _In_opt_ ID3D11DeviceContext * d3dContext,
            _In_reads_bytes_(dds_dataSize) const uint8_t * dds_data,
            _In_ size_t dds_dataSize,
            _In_ size_t maxsize,
            _In_ D3D11_USAGE usage,
            _In_ uint32_t bindFlags,
            _In_ uint32_t cpu_access_flags,
            _In_ uint32_t misc_flags,
            _In_ bool forceSRGB,
            _Outptr_opt_ ID3D11Resource ** texture,
            _Outptr_opt_ ID3D11ShaderResourceView ** texture_view,
            _Out_opt_ DDS_ALPHA_MODE * alpha_mode = nullptr )
```

4.1.3.29 CreateDDSTextureFromMemoryEx() [3/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromMemoryEx (
            ID3D11Device * d3dDevice,
            const uint8_t * dds_data,
            size_t dds_dataSize,
            size_t maxsize,
            D3D11_USAGE usage,
            uint32_t bindFlags,
            uint32_t cpu_access_flags,
            uint32_t misc_flags,
            bool forceSRGB,
            ID3D11Resource ** texture,
            ID3D11ShaderResourceView ** texture_view,
            DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.30 CreateDDSTextureFromMemoryEx() [4/4]

```
_Use_decl_annotations_ HRESULT mage::CreateDDSTextureFromMemoryEx (
             ID3D11Device * d3dDevice,
             ID3D11DeviceContext * d3dContext,
             const uint8_t * dds_data,
             size_t dds_dataSize,
             size_t maxsize,
             D3D11_USAGE usage,
             uint32_t bindFlags,
             uint32_t cpu_access_flags,
             uint32_t misc_flags,
             bool forceSRGB,
             ID3D11Resource ** texture,
             ID3D11ShaderResourceView ** texture_view,
             DDS_ALPHA_MODE * alpha_mode )
```

4.1.3.31 CreateTextureFromDDS()

```
static HRESULT mage::CreateTextureFromDDS (
             _In_ ID3D11Device * d3dDevice,
             _In_opt_ ID3D11DeviceContext * d3dContext,
             _In_ const DDS_HEADER * header,
             _In_reads_bytes_(bit_size) const uint8_t * bit_data,
             _In_ size_t bit_size,
             _In_ size_t maxsize,
             _In_ D3D11_USAGE usage,
             _In_ uint32_t bindFlags,
             _In_ uint32_t cpu_access_flags,
             _In_ uint32_t misc_flags,
             _In_ bool forceSRGB,
             _Outptr_opt_ ID3D11Resource ** texture,
             _Outptr_opt_ ID3D11ShaderResourceView ** texture_view )  [static]
```

4.1.3.32 Debug()

```
void mage::Debug (
             const char * format,
              ... )
```

Notifies a debug message.

A debug message is associated with generally useful information to log only in debug builds.

Parameters

in	*format*	Pointer to the message format.

4.1.3.33 EnqueueTasks()

```
void mage::EnqueueTasks (
             const vector< Task * > & tasks )
```

4.1.3.34 Error()

```
void mage::Error (
             const char * format,
              ... )
```

Notifies an error message.

An error message is associated with any error which is fatal to the operation, but not the service or application.

Parameters

in	*format*	Pointer to the message format.

4.1.3.35 Fatal()

```
void mage::Fatal (
             const char * format,
              ... )
```

Notifies a fatal message.

A fatal message is associated with any error that is forcing a shutdown of the service or application to prevent data loss (or further data loss).

Parameters

in	*format*	Pointer to the message format.

4.1.3.36 FillInitData()

```
static HRESULT mage::FillInitData (
             _In_ size_t width,
             _In_ size_t height,
             _In_ size_t depth,
             _In_ size_t mip_count,
             _In_ size_t array_size,
             _In_ DXGI_FORMAT format,
             _In_ size_t maxsize,
             _In_ size_t bit_size,
             _In_reads_bytes_(bit_size) const uint8_t * bit_data,
             _Out_ size_t & twidth,
             _Out_ size_t & theight,
             _Out_ size_t & tdepth,
             _Out_ size_t & skip_mip,
             _Out_writes_(mip_count *array_size) D3D11_SUBRESOURCE_DATA * init_data )  [static]
```

4.1.3.37 FindWordEnd()

```
static const char* mage::FindWordEnd (
            const char * buffer )  [static]
```

Finds the end of a word.

Parameters

in	*buffer*	Pointer to the first character.

Returns

Pointer to the end of the word. This means the pointer points to a space or null-terminating character.

4.1.3.38 FreeAligned()

```
void mage::FreeAligned (
            void * ptr )
```

Frees a block of memory that was allocated with mage::AllocAligned(size_t) or mage::AllocAligned<T>(size_t).

Parameters

in	*ptr*	A pointer to the memory block that was allocated.

4.1.3.39 GetAlphaMode()

```
static DDS_ALPHA_MODE mage::GetAlphaMode (
            _In_ const DDS_HEADER * header )  [static]
```

4.1.3.40 GetDXGIFormat()

```
static DXGI_FORMAT mage::GetDXGIFormat (
            const DDS_PIXELFORMAT & ddpf )  [static]
```

4.1.3.41 GetSurfaceInfo()

```
static void mage::GetSurfaceInfo (
             _In_ size_t width,
             _In_ size_t height,
             _In_ DXGI_FORMAT fmt,
             _Out_opt_ size_t * out_nb_bytes,
             _Out_opt_ size_t * out_row_bytes,
             _Out_opt_ size_t * out_nb_rows )  [static]
```

4.1.3.42 Info()

```
void mage::Info (
             const char * format,
              ...  )
```

Notifies an info message.

An info message is associated with generally useful information to log.

Parameters

in	*format*	Pointer to the message format.

4.1.3.43 InitializeConsole()

```
HRESULT mage::InitializeConsole ( )
```

Allocates a console to this engine for basic io and redirects stdin, stdout and stderr to the allocated console.

Returns

A success/error value.

4.1.3.44 LoadTextureDataFromFile()

```
static HRESULT mage::LoadTextureDataFromFile (
             _In_z_ const wchar_t * file_name,
             std::unique_ptr< uint8_t[]> & dds_data,
             DDS_HEADER ** header,
             uint8_t ** bit_data,
             size_t * bit_size )  [static]
```

4.1.3.45 MainWindowProc()

```
LRESULT CALLBACK mage::MainWindowProc (
             HWND hWnd,
             UINT msg,
             WPARAM wParam,
             LPARAM lParam )
```

The application-defined function that processes messages sent to the engine window. The WindowProc type defines a pointer to this callback function.

Parameters

in	*hWnd*	A handle to the window.
in	*msg*	The message.
in	*wParam*	Additional message information. The contents of this parameter depend on the value of *msg*.
in	*lParam*	Additional message information. The contents of this parameter depend on the value of *msg*.

Returns

The return value is the result of the message processing and depends on the message sent.

4.1.3.46 MakeSRGB()

```
static DXGI_FORMAT mage::MakeSRGB (
             _In_ DXGI_FORMAT format )  [static]
```

4.1.3.47 MaxThreadIndex()

```
size_t mage::MaxThreadIndex ( )
```

4.1.3.48 NumberOfSystemCores()

```
size_t mage::NumberOfSystemCores ( )
```

Returns the number of system cores (i.e. logical processors).

Returns

The number of system cores (i.e. logical processors).

4.1.3.49 Overlap()

```
AABB mage::Overlap (
             const AABB & aabb1,
             const AABB & aabb2 )
```

Returns the overlap AABB of the two given AABBs.

Parameters

in	*aabb1*	A reference to the first AABB.
in	*aabb2*	A reference to the second AABB.

Returns

The identity AABB in case of no overlap.
The overlap AABB of *aabb1* and *aabb2*.

4.1.3.50 OverlapStrict()

```
AABB mage::OverlapStrict (
             const AABB & aabb1,
             const AABB & aabb2 )
```

Returns the strict overlap AABB of the two given AABBs.

Parameters

in	*aabb1*	A reference to the first AABB.
in	*aabb2*	A reference to the second AABB.

Returns

The identity AABB in case of no strict overlap.
The strict overlap AABB of *aabb1* and *aabb2*.

4.1.3.51 ParallelCleanup()

```
void mage::ParallelCleanup ( )
```

4.1.3.52 ParallelFor()

```
void mage::ParallelFor (
             function< void(size_t) > func,
             size_t nb_work,
             size_t chunk_size = 1 )
```

4.1.3.53 ParallelInit()

```
void mage::ParallelInit ( )
```

4.1.3.54 PrintConsoleHeader()

```
void mage::PrintConsoleHeader ( )
```

Prints the header of the engine to the console.

4.1.3.55 ProcessError()

```
static void mage::ProcessError (
             const char * format,
             const va_list args,
             const string & error_type,
             int error_disposition )  [static]
```

Process the given error.

Parameters

in	*format*	The format of the error string.
in	*args*	The arguments of the format string.
in	*error_type*	The type of the error.
in	*error_disposition*	Disposition of the error.

4.1.3.56 RejectDisplayMode()

```
bool mage::RejectDisplayMode (
            const DXGI_MODE_DESC1 * display_mode_desc )
```

Checks whether the given display mode needs to be rejected for the engine.

Parameters

in	*display_mode_desc*	A pointer to a display mode descriptor.

Returns

`true` if the given display mode needs to be rejected for the engine. `false` otherwise.

4.1.3.57 SettingsDialogProcDelegate()

```
INT_PTR CALLBACK mage::SettingsDialogProcDelegate (
            HWND hwndDlg,
            UINT uMsg,
            WPARAM wParam,
            LPARAM lParam )
```

Engine-defined callback function used with the CreateDialog for device enumeration.

Parameters

in	*hwndDlg*	A handle to the dialog box.
in	*uMsg*	The message.
in	*wParam*	Additional message-specific information.
in	*lParam*	Additional message-specific information.

Returns

`true` if *uMsg* is processed. `false` otherwise.

4.1.3.58 task_entry()

```
static DWORD WINAPI mage::task_entry (
            LPVOID lpParameter ) [static]
```

An application-defined function that serves as the starting address for a thread.

Parameters

in	*lpParameter*	The thread data passed to the function using the `lpParameter` parameter of CreateThread.

Returns

A value indicating success or failure.

4.1.3.59 TasksCleanup()

```
void mage::TasksCleanup ( )
```

4.1.3.60 TasksInit()

```
void mage::TasksInit ( )
```

4.1.3.61 TerminalWidth()

```
int mage::TerminalWidth ( )
```

Returns the fixed terminal width.

Returns

The fixed terminal width.

4.1.3.62 Union() [1/2]

```
AABB mage::Union (
             const AABB & aabb,
             const XMFLOAT3 & point )
```

Returns the union AABB of the given AABB and the given point.

Parameters

in	*aabb*	A reference to the AABB.
in	*point*	A reference to the point.

Returns

The union AABB of *aabb* and *point*.

4.1.3.63 Union() [2/2]

```
AABB mage::Union (
             const AABB & aabb1,
             const AABB & aabb2 )
```

Returns the union AABB of the two given AABBs.

Parameters

in	*aabb1*	A reference to the first AABB.
in	*aabb2*	A reference to the second AABB.

Returns

The union AABB of *aabb1* and *aabb2*.

4.1.3.64 WaitForAllTasks()

```
void mage::WaitForAllTasks ( )
```

4.1.3.65 Warning()

```
void mage::Warning (
             const char * format,
              ... )
```

Notifies a warning message.

A warning message is associated with anything that can potentially cause application oddities.

Parameters

in	*format*	Pointer to the message format.

4.1.4 Variable Documentation

4.1.4.1 DDS_MAGIC

```
const uint32_t mage::DDS_MAGIC = 0x20534444
```

4.1.4.2 g_device_enumeration

```
DeviceEnumeration * mage::g_device_enumeration = nullptr
```

A (global) pointer to the device enumeration.

4.1.4.3 g_engine

```
Engine * mage::g_engine = nullptr
```

The engine used by the user.

4.1.4.4 g_feature_levels

```
const D3D_FEATURE_LEVEL mage::g_feature_levels[ ]
```

Initial value:

```
= {
        D3D_FEATURE_LEVEL_11_1,
        D3D_FEATURE_LEVEL_11_0
    }
```

The supported feature levels.

4.1.4.5 g_logging_configuration

```
LoggingConfiguration mage::g_logging_configuration
```

The logging configuration defined by the user and used by the engine.

4.1.4.6 g_pixel_formats

```
const DXGI_FORMAT mage::g_pixel_formats[ ]
```

Initial value:

```
= {
        DXGI_FORMAT_B5G5R5A1_UNORM,
        DXGI_FORMAT_B5G6R5_UNORM,
        DXGI_FORMAT_B8G8R8X8_UNORM,
        DXGI_FORMAT_B8G8R8A8_UNORM,
        DXGI_FORMAT_R10G10B10A2_UNORM,
    }
```

The allowed pixel formats.

4.1.4.7 lvertex_input_element_desc

```
const D3D11_INPUT_ELEMENT_DESC mage::lvertex_input_element_desc[ ]
```

Initial value:

```
= {
        { "POSITION", 0, DXGI_FORMAT_R32G32B32_FLOAT,    0, UINT(offsetof(LVertex, p)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 },
        { "DIFFUSE",  0, DXGI_FORMAT_R32G32B32A32_FLOAT, 0, UINT(offsetof(LVertex, diffuse)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 },
        { "TEXCOORD", 0, DXGI_FORMAT_R32G32_FLOAT,       0, UINT(offsetof(LVertex, tex)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 }
    }
```

The input element descriptor for a LVertex.

4.1.4.8 nb_unfinished_tasks

```
uint32_t mage::nb_unfinished_tasks  [static]
```

The number of unfinished tasks.

4.1.4.9 shutdown_workers

```
bool mage::shutdown_workers = false  [static]
```

4.1.4.10 task_queue

```
vector< Task * > mage::task_queue  [static]
```

The task queue.

4.1.4.11 task_queue_mutex

```
Mutex* mage::task_queue_mutex = Mutex::Create()  [static]
```

The mutex for exclusive access to the task queue.

4.1.4.12 tasks_running_condition

```
ConditionVariable* mage::tasks_running_condition  [static]
```

The running condition variable for exclusive access to the number of unfinished tasks and for signaling on updates.

4.1.4.13 ThreadIndex

```
thread_local size_t mage::ThreadIndex
```

4.1.4.14 threads

```
HANDLE* mage::threads  [static]
```

The thread handles.

4.1.4.15 tlvertex_input_element_desc

```
const D3D11_INPUT_ELEMENT_DESC mage::tlvertex_input_element_desc[ ]
```

Initial value:

```
= {
        { "POSITION", 0, DXGI_FORMAT_R32G32B32A32_FLOAT, 0, UINT(offsetof(TLVertex, p)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 },
        { "DIFFUSE",  0, DXGI_FORMAT_R32G32B32A32_FLOAT, 0, UINT(offsetof(TLVertex, diffuse)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 },
        { "TEXCOORD", 0, DXGI_FORMAT_R32G32_FLOAT,       0, UINT(offsetof(TLVertex, tex)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 }
    }
```

The input element descriptor for a TLVertex

4.1.4.16 vertex_input_element_desc

```
const D3D11_INPUT_ELEMENT_DESC mage::vertex_input_element_desc[ ]
```

Initial value:

```
= {
        { "POSITION", 0, DXGI_FORMAT_R32G32B32_FLOAT, 0, UINT(offsetof(Vertex, p)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 },
        { "NORMAL",   0, DXGI_FORMAT_R32G32B32_FLOAT, 0, UINT(offsetof(Vertex, n)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 },
        { "TEXCOORD", 0, DXGI_FORMAT_R32G32_FLOAT,    0, UINT(offsetof(Vertex, tex)),
      D3D11_INPUT_PER_VERTEX_DATA, 0 }
    }
```

The input element descriptor for a Vertex.

4.1.4.17 work_list

```
ParallelForLoop* mage::work_list = nullptr  [static]
```

4.1.4.18 work_list_condition

```
condition_variable mage::work_list_condition  [static]
```

4.1.4.19 work_list_mutex

```
mutex mage::work_list_mutex  [static]
```

4.1.4.20 worker_semaphore

```
Semaphore* mage::worker_semaphore  [static]
```

The worker semaphore for being able to work.

4.1.4.21 workers

```
vector< thread > mage::workers  [static]
```

Chapter 5

Class Documentation

5.1 mage::AABB Struct Reference

```
#include <bounding_volume.hpp>
```

Public Member Functions

- AABB ()
- AABB (const XMFLOAT3 &p_min, const XMFLOAT3 &p_max)
- bool Encloses (const AABB &aabb) const
- bool EnclosesStrict (const AABB &aabb) const
- bool Encloses (const XMFLOAT3 &point) const
- bool EnclosesStrict (const XMFLOAT3 &point) const
- bool Encloses (const Face &face) const
- bool EnclosesStrict (const Face &face) const
- bool EnclosedBy (const list< XMFLOAT4 > &planes) const
- bool EnclosedStrictBy (const list< XMFLOAT4 > &planes) const
- bool Overlaps (const AABB &aabb) const
- bool OverlapsStrict (const AABB &aabb) const
- XMFLOAT3 Diagonal () const

Public Attributes

- const XMFLOAT3 p_min
- const XMFLOAT3 p_max

5.1.1 Detailed Description

A struct of Axis-Aligned Bounding Boxes (AABBs).

5.1.2 Constructor & Destructor Documentation

5.1.2.1 AABB() [1/2]

```
mage::AABB::AABB ( )
```

Constructs an (identity) AABB.

5.1.2.2 AABB() [2/2]

```
mage::AABB::AABB (
            const XMFLOAT3 & p_min,
            const XMFLOAT3 & p_max )
```

Constructs an AABB.

Parameters

in	*p_min*	The minimum extents.
in	*p_max*	The maximum extents.

5.1.3 Member Function Documentation

5.1.3.1 Diagonal()

```
XMFLOAT3 mage::AABB::Diagonal ( ) const
```

Returns the diagonal of this AABB.

Returns

The diagonal of this AABB.

5.1.3.2 EnclosedBy()

```
bool mage::AABB::EnclosedBy (
            const list< XMFLOAT4 > & planes ) const
```

Checks whether this AABB is completely enclosed by the given (closed) volume.

Parameters

in	*planes*	A reference to a linked list containing the planes of the volume (each plane's coefficients are represented as a XMFLOAT4).

Returns

> `true` if this AABB is completely enclosed by *planes*. `false` otherwise.

5.1.3.3 EnclosedStrictBy()

```
bool mage::AABB::EnclosedStrictBy (
            const list< XMFLOAT4 > & planes ) const
```

Checks whether this AABB is completely, strictly enclosed by the given (closed) volume.

Parameters

in	*planes*	A reference to a linked list containing the planes of the volume (each plane's coefficients are represented as a `XMFLOAT4`).

Returns

> `true` if this AABB is completely, stricly enclosed by *planes*. `false` otherwise.

5.1.3.4 Encloses() [1/3]

```
bool mage::AABB::Encloses (
            const AABB & aabb ) const
```

Checks whether this AABB completely encloses the given AABB.

Parameters

in	*aabb*	A reference to the AABB.

Returns

> `true` if this AABB completely encloses *aabb*. `false` otherwise.

5.1.3.5 Encloses() [2/3]

```
bool mage::AABB::Encloses (
            const XMFLOAT3 & point ) const
```

Checks whether this AABB completely encloses the given point.

Parameters

in	*point*	A reference to the point.

Returns

`true` if this AABB completely encloses *point*. `false` otherwise.

5.1.3.6 Encloses() [3/3]

```
bool mage::AABB::Encloses (
             const Face & face ) const
```

Checks whether this AABB completely encloses the given face.

Parameters

in	*face*	A reference to the face.

Returns

`true` if this AABB completely encloses *face*. `false` otherwise.

5.1.3.7 EnclosesStrict() [1/3]

```
bool mage::AABB::EnclosesStrict (
             const AABB & aabb ) const
```

Checks whether this AABB completely, strictly encloses the given AABB.

Parameters

in	*aabb*	A reference to the AABB.

Returns

`true` if this AABB completely, strictly encloses *aabb*. `false` otherwise.

5.1.3.8 EnclosesStrict() [2/3]

```
bool mage::AABB::EnclosesStrict (
             const XMFLOAT3 & point ) const
```

Checks whether this AABB completely, strictly encloses the given point.

Parameters

in	*point*	A reference to the point.

Returns

`true` if this AABB completely, strictly encloses *point*. `false` otherwise.

5.1.3.9 EnclosesStrict() [3/3]

```
bool mage::AABB::EnclosesStrict (
            const Face & face ) const
```

Checks whether this AABB completely, strictly encloses the given face.

Parameters

in	*face*	A reference to the face.

Returns

`true` if this AABB completely, strictly encloses *face*. `false` otherwise.

5.1.3.10 Overlaps()

```
bool mage::AABB::Overlaps (
            const AABB & aabb ) const
```

Checks whether this AABB overlaps the given AABB.

Parameters

in	*aabb*	A reference to the AABB.

Returns

`true` if this AABB overlaps *aabb*. `false` otherwise.

5.1.3.11 OverlapsStrict()

```
bool mage::AABB::OverlapsStrict (
            const AABB & aabb ) const
```

Checks whether this AABB strictly overlaps the given AABB.

Parameters

in	*aabb*	A reference to the AABB.

Returns

> `true` if this AABB strictly overlaps *aabb*. `false` otherwise.

5.1.4 Member Data Documentation

5.1.4.1 p_max

```
const XMFLOAT3 mage::AABB::p_max
```

The maximum extents of this AABB.

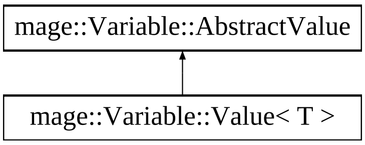
5.1.4.2 p_min

```
const XMFLOAT3 mage::AABB::p_min
```

The minimum extents of this AABB.

5.2 mage::Variable::AbstractValue Struct Reference

Inheritance diagram for mage::Variable::AbstractValue:



Public Member Functions

- virtual ∼AbstractValue ()
- virtual const void ∗ GetValue () const =0

5.2.1 Detailed Description

A struct of immutable abstract values.

Note

> This is an example of the Type Erasure pattern for templates. We need to keep the original type to ensure the right destructor can be called in case of non-primitive types.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 ∼AbstractValue()

```
virtual mage::Variable::AbstractValue::∼AbstractValue ( )  [virtual]
```

Destructs this value.

5.2.3 Member Function Documentation

5.2.3.1 GetValue()

```
virtual const void* mage::Variable::AbstractValue::GetValue ( ) const  [pure virtual]
```

Returns the value of this value.

Returns

A pointer to the value of this value.

Implemented in mage::Variable::Value< T >.

5.3 mage::BS Struct Reference

```
#include <bounding_volume.hpp>
```

Public Member Functions

- BS ()
- BS (const XMFLOAT3 &p, float r)
- bool Encloses (const list< XMFLOAT4 > &planes)
- bool EnclosesStrict (const list< XMFLOAT4 > &planes)
- bool Collides (const BS &sphere, const XMFLOAT3 velocity_sum, float *collision_distance)

Public Attributes

- XMFLOAT3 p
- float r

5.3.1 Detailed Description

A struct of Bounding Spheres (BS).

5.3.2 Constructor & Destructor Documentation

5.3.2.1 BS() [1/2]

```
mage::BS::BS ( )
```

Constructs a sphere.

5.3.2.2 BS() [2/2]

```
mage::BS::BS (
             const XMFLOAT3 & p,
             float r )
```

Constructs a sphere.

Parameters

in	*p*	The position
in	*r*	The radius.

5.3.3 Member Function Documentation

5.3.3.1 Collides()

```
bool mage::BS::Collides (
            const BS & sphere,
            const XMFLOAT3 velocity_sum,
            float * collision_distance )
```

Checks whether this sphere collides with a given sphere.

Parameters

in	*sphere*	The sphere.
in	*velocity_sum*	The sum of the velocities of both spheres.
out	*collision_distance*	The collision distance (in case of collision).

Returns

`true` if this sphere collides with *sphere*. `false` otherwise.

5.3.3.2 Encloses()

```
bool mage::BS::Encloses (
            const list< XMFLOAT4 > & planes )
```

Checks whether this sphere completely encloses the given (closed) volume.

Parameters

in	*planes*	A reference to a linked list containing the planes of the volume (each plane's coefficients are represented as a `XMFLOAT4`).

Returns

`true` if this sphere completely encloses *planes*. `false` otherwise.

5.3.3.3 EnclosesStrict()

```
bool mage::BS::EnclosesStrict (
            const list< XMFLOAT4 > & planes )
```

Checks whether this sphere completely, strictly encloses the given (closed) volume.

Parameters

in	*planes*	A reference to a linked list containing the planes of the volume (each plane's coefficients are represented as a `XMFLOAT4`).

Returns

`true` if this sphere completely encloses *planes*. `false` otherwise.

5.3.4 Member Data Documentation

5.3.4.1 p

```
XMFLOAT3 mage::BS::p
```

The position of this sphere.

5.3.4.2 r

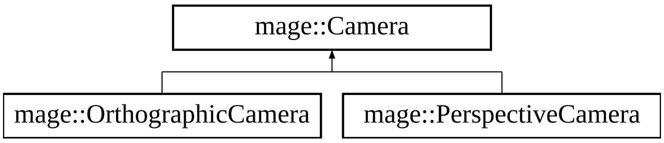
```
float mage::BS::r
```

The radius of this sphere.

5.4 mage::Camera Class Reference

```
#include <camera.hpp>
```

Inheritance diagram for mage::Camera:



Public Member Functions

- virtual ~Camera ()
- float GetWidth () const
- Camera & SetWidth (float width)
- float GetHeight () const
- Camera & SetHeight (float height)
- Camera & SetWidthAndHeight (float width, float height)
- float GetNearZ () const
- Camera & SetNearZ (float near_z)
- float GetFarZ () const
- Camera & SetFarZ (float far_z)
- Camera & SetNearAndFarZ (float near_z, float far_z)
- virtual XMMATRIX GetViewToProjectionMatrix () const =0

Protected Member Functions

- Camera (float width, float height, float near_z=MAGE_DEFAULT_CAMERA_NEAR_Z, float far_z=MAGE_DEFAULT_CAMERA_FAR_Z)

Protected Attributes

- float m_width
- float m_height
- float m_near_z
- float m_far_z

5.4.1 Detailed Description

A class of camera.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 ∼Camera()

```
virtual mage::Camera::∼Camera ( )  [virtual]
```

Destructs this camera.

5.4.2.2 Camera()

```
mage::Camera::Camera (
             float width,
             float height,
             float near_z = MAGE_DEFAULT_CAMERA_NEAR_Z,
             float far_z = MAGE_DEFAULT_CAMERA_FAR_Z )  [protected]
```

Constructs a camera.

Parameters

in	*width*	The width.
in	*height*	The height.
in	*near_z*	The position of the near z-plane.
in	*far_z*	The position of the far z-plane.

5.4.3 Member Function Documentation

5.4.3.1 GetFarZ()

```
float mage::Camera::GetFarZ ( ) const
```

Returns the position of the far z-plane of this camera.

Returns

The position of the far z-plane of this camera.

5.4.3.2 GetHeight()

```
float mage::Camera::GetHeight ( ) const
```

Returns the height of this camera.

Returns

The height of this camera.

5.4.3.3 GetNearZ()

```
float mage::Camera::GetNearZ ( ) const
```

Returns the position of the near z-plane of this camera.

Returns

The position of the near z-plane of this camera.

5.4.3.4 GetViewToProjectionMatrix()

```
virtual XMMATRIX mage::Camera::GetViewToProjectionMatrix ( ) const  [pure virtual]
```

Returns the view-to-projection matrix of this camera.

Returns

The view-to-projection matrix of this camera.

Implemented in mage::PerspectiveCamera, and mage::OrthographicCamera.

5.4.3.5 GetWidth()

```
float mage::Camera::GetWidth ( ) const
```

Returns the width of this camera.

Returns

The width of this camera.

5.4.3.6 SetFarZ()

```
Camera& mage::Camera::SetFarZ (
             float far_z )
```

Sets the position of the far z-plane of this camera to the given value.

Parameters

in	*far_z*	The position of the far z-plane.

Returns

A reference to this camera.

5.4.3.7 SetHeight()

```
Camera& mage::Camera::SetHeight (
            float height )
```

Sets the height of this camera to the given value.

Parameters

in	*height*	The height.

Returns

A reference to this camera.

5.4.3.8 SetNearAndFarZ()

```
Camera& mage::Camera::SetNearAndFarZ (
            float near_z,
            float far_z )
```

Sets the position of the near and far z-plane of this camera to the given values.

Parameters

in	*near_z*	The position of the near z-plane.
in	*far_z*	The position of the far z-plane.

Returns

A reference to this camera.

5.4.3.9 SetNearZ()

```
Camera& mage::Camera::SetNearZ (
            float near_z )
```

Sets the position of the near z-plane of this camera to the given value.

Parameters

in	*near↩ _z*	The position of the near z-plane.

Returns

A reference to this camera.

5.4.3.10 SetWidth()

```
Camera& mage::Camera::SetWidth (
             float width )
```

Sets the width of this camera to the given value.

Parameters

in	*width*	The width.

Returns

A reference to this camera.

5.4.3.11 SetWidthAndHeight()

```
Camera& mage::Camera::SetWidthAndHeight (
             float width,
             float height )
```

Sets the width and height of this camera to the given values.

Parameters

in	*width*	The width.
in	*height*	The height.

Returns

A reference to this camera.

5.4.4 Member Data Documentation

5.4.4.1 m_far_z

```
float mage::Camera::m_far_z  [protected]
```

The position of the far z-plane.

5.4.4.2 m_height

```
float mage::Camera::m_height  [protected]
```

The height of this camera.

5.4.4.3 m_near_z

```
float mage::Camera::m_near_z  [protected]
```

The position of the near z-plane.

5.4.4.4 m_width

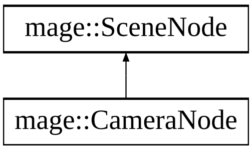
```
float mage::Camera::m_width  [protected]
```

The width of this camera.

5.5 mage::CameraNode Class Reference

```
#include <camera_node.hpp>
```

Inheritance diagram for mage::CameraNode:



Public Member Functions

- CameraNode (Camera ∗camera, const Transform &transform=Transform())
- virtual ∼CameraNode ()
- Camera ∗ GetCamera () const
- virtual void Accept (SceneNodeVisitor &visitor) override
- virtual void Accept (SceneNodeVisitor &visitor) const override

Protected Attributes

- Camera ∗ m_camera

Additional Inherited Members

5.5.1 Detailed Description

A class of camera nodes.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CameraNode()

```
mage::CameraNode::CameraNode (
            Camera * camera,
            const Transform & transform = Transform() )
```

Constructs a camera node with given camera and transform.

Precondition

camera may not point to `nullptr`.

Parameters

in	*camera*	A pointer to the camera.
in	*transform*	A reference to the transform.

5.5.2.2 ∼CameraNode()

```
virtual mage::CameraNode::∼CameraNode ( )  [virtual]
```

Destructs this camera node.

5.5.3 Member Function Documentation

5.5.3.1 Accept() [1/2]

```
virtual void mage::CameraNode::Accept (
            SceneNodeVisitor & visitor )  [override], [virtual]
```

Accepts the given visitor.

Parameters

in	*visitor*	A reference to the visitor.

Implements mage::SceneNode.

5.5.3.2 Accept() [2/2]

```
virtual void mage::CameraNode::Accept (
            SceneNodeVisitor & visitor ) const  [override], [virtual]
```

Accepts the given visitor.

Parameters

in	*visitor*	A reference to the visitor.

Implements mage::SceneNode.

5.5.3.3 GetCamera()

```
Camera* mage::CameraNode::GetCamera ( ) const
```

Returns the camera of this camera node.

Returns

A pointer to the camera of this camera node.

5.5.4 Member Data Documentation

5.5.4.1 m_camera

```
Camera* mage::CameraNode::m_camera  [protected]
```

A pointer to the camera of this camera node.

5.6 mage::CartesianAxesSystem Struct Reference

```
#include <coordinate_system.hpp>
```

Public Member Functions

- CartesianAxesSystem ()
- CartesianAxesSystem (const XMVECTOR &x)
- CartesianAxesSystem (const XMVECTOR &x, const XMVECTOR &y)
- CartesianAxesSystem (const XMVECTOR &x, const XMVECTOR &y, const XMVECTOR &z)
- CartesianAxesSystem (const CartesianAxesSystem &axes)
- ∼CartesianAxesSystem ()
- CartesianAxesSystem & operator= (const CartesianAxesSystem &axes)
- XMVECTOR GetAxisX () const
- XMVECTOR GetAxisY () const
- XMVECTOR GetAxisZ () const

Private Attributes

- XMVECTOR m_x
- XMVECTOR m_y
- XMVECTOR m_z

5.6.1 Detailed Description

A struct of Cartesian axes systems.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 CartesianAxesSystem() [1/5]

```
mage::CartesianAxesSystem::CartesianAxesSystem ( )
```

Constructs a Cartesian axes system.

5.6.2.2 CartesianAxesSystem() [2/5]

```
mage::CartesianAxesSystem::CartesianAxesSystem (
            const XMVECTOR & x )
```

Constructs a Cartesian axes system from the given axes.

Precondition

The given axis is normalized.

Parameters

in	*x*	The x-axis.

5.6.2.3 CartesianAxesSystem() [3/5]

```
mage::CartesianAxesSystem::CartesianAxesSystem (
            const XMVECTOR & x,
            const XMVECTOR & y )
```

Constructs a Cartesian axes system from the given axes.

Precondition

The given axes are orthonormal.

Parameters

in	*x*	The x-axis.
in	*y*	The y-axis.

5.6.2.4 CartesianAxesSystem() [4/5]

```
mage::CartesianAxesSystem::CartesianAxesSystem (
            const XMVECTOR & x,
            const XMVECTOR & y,
            const XMVECTOR & z )
```

Constructs a Cartesian axes system from the given axes.

Precondition

The given axes are orthonormal.

Parameters

in	*x*	The x-axis.
in	*y*	The y-axis.
in	*z*	The z-axis.

5.6.2.5 CartesianAxesSystem() [5/5]

```
mage::CartesianAxesSystem::CartesianAxesSystem (
            const CartesianAxesSystem & axes )
```

Constructs a Cartesian axes system from the given Cartesian axes system.

Parameters

in	*axes*	The Cartesian axes system.

5.6.2.6 ∼CartesianAxesSystem()

```
mage::CartesianAxesSystem::∼CartesianAxesSystem ( )
```

Destructs this Cartesian axes system.

5.6.3 Member Function Documentation

5.6.3.1 GetAxisX()

```
XMVECTOR mage::CartesianAxesSystem::GetAxisX ( ) const
```

Returns the x-axis of this Cartesian axes system.

Returns

The x-axis of this Cartesian axes system.

5.6.3.2 GetAxisY()

```
XMVECTOR mage::CartesianAxesSystem::GetAxisY ( ) const
```

Returns the y-axis of this Cartesian axes system.

Returns

The y-axis of this Cartesian axes system.

5.6.3.3 GetAxisZ()

```
XMVECTOR mage::CartesianAxesSystem::GetAxisZ ( ) const
```

Returns the z-axis of this Cartesian axes system.

Returns

The z-axis of this Cartesian axes system.

5.6.3.4 operator=()

```
CartesianAxesSystem& mage::CartesianAxesSystem::operator= (
             const CartesianAxesSystem & axes )
```

Copies the given Cartesian axes system to this Cartesian axes system.

Parameters

in	*axes*	The Cartesian axes system to copy from.

Returns

A reference to the copy of the given Cartesian axes system (i.e. this Cartesian axes system).

5.6.4 Member Data Documentation

5.6.4.1 m_x

```
XMVECTOR mage::CartesianAxesSystem::m_x [private]
```

The x-axis of this Cartesian axes system.

5.6.4.2 m_y

```
XMVECTOR mage::CartesianAxesSystem::m_y [private]
```

The y-axis of this Cartesian axes system.

5.6.4.3 m_z

```
XMVECTOR mage::CartesianAxesSystem::m_z  [private]
```

The z-axis of this Cartesian axes system.

5.7 mage::CartesianCoordinateSystem Struct Reference

```
#include <coordinate_system.hpp>
```

Public Member Functions

- CartesianCoordinateSystem (const CartesianAxesSystem &axes)
- CartesianCoordinateSystem (const XMVECTOR &o, const CartesianAxesSystem &axes)
- CartesianCoordinateSystem (const CartesianCoordinateSystem &coordinate_system)
- ∼CartesianCoordinateSystem ()
- CartesianCoordinateSystem & operator= (const CartesianCoordinateSystem &coordinate_system)
- XMVECTOR GetOrigin () const
- XMVECTOR GetAxisX () const
- XMVECTOR GetAxisY () const
- XMVECTOR GetAxisZ () const
- CartesianAxesSystem GetAxes () const

Private Attributes

- XMVECTOR m_o
- CartesianAxesSystem m_axes

5.7.1 Detailed Description

A struct of Cartesian coordinate systems.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 CartesianCoordinateSystem() [1/3]

```
mage::CartesianCoordinateSystem::CartesianCoordinateSystem (
             const CartesianAxesSystem & axes )  [explicit]
```

Constructs a Cartesian coordinate system from the given Cartesian axes system.

Parameters

in	*axes*	The Cartesian axes system.

5.7.2.2 CartesianCoordinateSystem() [2/3]

```
mage::CartesianCoordinateSystem::CartesianCoordinateSystem (
            const XMVECTOR & o,
            const CartesianAxesSystem & axes )
```

Constructs a Cartesian coordinate system from the given origin and Cartesian axes system.

Parameters

in	*o*	The origin.
in	*axes*	The Cartesian axes system.

5.7.2.3 CartesianCoordinateSystem() [3/3]

```
mage::CartesianCoordinateSystem::CartesianCoordinateSystem (
            const CartesianCoordinateSystem & coordinate_system )
```

Constructs a Cartesian coordinate system from the given Cartesian coordinate system.

Parameters

in	*coordinate_system*	The Cartesian coordinate system.

5.7.2.4 ∼CartesianCoordinateSystem()

```
mage::CartesianCoordinateSystem::∼CartesianCoordinateSystem ( )
```

Destructs this Cartesian coordinate system.

5.7.3 Member Function Documentation

5.7.3.1 GetAxes()

```
CartesianAxesSystem mage::CartesianCoordinateSystem::GetAxes ( ) const
```

Returns the axes of this Cartesian coordinate system.

Returns

The Cartesian axes system of this Cartesian coordinate system.

5.7.3.2 GetAxisX()

```
XMVECTOR mage::CartesianCoordinateSystem::GetAxisX ( ) const
```

Returns the x-axis of this Cartesian coordinate system.

Returns

The x-axis of this Cartesian coordinate system.

5.7.3.3 GetAxisY()

```
XMVECTOR mage::CartesianCoordinateSystem::GetAxisY ( ) const
```

Returns the y-axis of this Cartesian coordinate system.

Returns

The y-axis of this Cartesian coordinate system.

5.7.3.4 GetAxisZ()

```
XMVECTOR mage::CartesianCoordinateSystem::GetAxisZ ( ) const
```

Returns the z-axis of this Cartesian coordinate system.

Returns

The z-axis of this Cartesian coordinate system.

5.7.3.5 GetOrigin()

```
XMVECTOR mage::CartesianCoordinateSystem::GetOrigin ( ) const
```

Returns the origin of this Cartesian coordinate system.

Returns

The origin of this Cartesian coordinate system.

5.7.3.6 operator=()

```
CartesianCoordinateSystem& mage::CartesianCoordinateSystem::operator= (
             const CartesianCoordinateSystem & coordinate_system )
```

Copies the given Cartesian coordinate system to this Cartesian coordinate system.

Parameters

in	*coordinate_system*	The Cartesian coordinate system to copy from.

Returns

A reference to the copy of the given Cartesian coordinate system (i.e. this Cartesian coordinate system).

5.7.4 Member Data Documentation

5.7.4.1 m_axes

```
CartesianAxesSystem mage::CartesianCoordinateSystem::m_axes  [private]
```

The Cartesian axes system of this Cartesian coordinate system.

5.7.4.2 m_o

```
XMVECTOR mage::CartesianCoordinateSystem::m_o  [private]
```

The origin of this Cartesian coordinate system.

5.8 mage::ConditionVariable Class Reference

```
#include <lock.hpp>
```

Public Member Functions

- ConditionVariable ()
- ∼ConditionVariable ()
- void Lock ()
- void Unlock ()
- void Wait ()
- void Signal ()

Private Types

- enum { SIGNAL = 0, BROADCAST = 1, NUM_EVENTS = 2 }

Private Attributes

- uint32_t m_nb_waiters
- CRITICAL_SECTION m_nb_waiters_mutex
- CRITICAL_SECTION m_condition_mutex
- HANDLE m_events [NUM_EVENTS]

5.8.1 Detailed Description

A class of condition variables.

5.8.2 Member Enumeration Documentation

5.8.2.1 anonymous enum

```
anonymous enum  [private]
```

Type of events (indices).

Enumerator

SIGNAL	
BROADCAST	
NUM_EVENTS	

5.8.3 Constructor & Destructor Documentation

5.8.3.1 ConditionVariable()

```
mage::ConditionVariable::ConditionVariable ( )
```

Constructs a condition variable.

5.8.3.2 ∼ConditionVariable()

```
mage::ConditionVariable::∼ConditionVariable ( )
```

Destructs this condition variable.

5.8.4 Member Function Documentation

5.8.4.1 Lock()

```
void mage::ConditionVariable::Lock ( )
```

Locks this condition variable.

5.8.4.2 Signal()

```
void mage::ConditionVariable::Signal ( )
```

Signal a condition change.

5.8.4.3 Unlock()

```
void mage::ConditionVariable::Unlock ( )
```

Unlocks this condition variable.

5.8.4.4 Wait()

```
void mage::ConditionVariable::Wait ( )
```

Wait for a signal indicating a condition change.

5.8.5 Member Data Documentation

5.8.5.1 m_condition_mutex

```
CRITICAL_SECTION mage::ConditionVariable::m_condition_mutex  [private]
```

The critical section object for the mutex guarding the condition of this condition variable.

5.8.5.2 m_events

```
HANDLE mage::ConditionVariable::m_events[NUM_EVENTS]  [private]
```

Signal and broadcast event handles of this condition variable.

5.8.5.3 m_nb_waiters

```
uint32_t mage::ConditionVariable::m_nb_waiters  [private]
```

The number of waiters of this condition variable.

5.8.5.4 m_nb_waiters_mutex

```
CRITICAL_SECTION mage::ConditionVariable::m_nb_waiters_mutex  [private]
```

The critical section object for the mutex guarding `m_nb_waiters` of this condition variable.

5.9 mage::DDS_HEADER Struct Reference

Public Attributes

- uint32_t size
- uint32_t flags
- uint32_t height
- uint32_t width
- uint32_t pitch_or_linear_size
- uint32_t depth
- uint32_t mip_map_count
- uint32_t reserved1 [11]
- DDS_PIXELFORMAT ddspf
- uint32_t caps
- uint32_t caps2
- uint32_t caps3
- uint32_t caps4
- uint32_t reserved2

5.9.1 Member Data Documentation

5.9.1.1 caps

```
uint32_t mage::DDS_HEADER::caps
```

5.9.1.2 caps2

```
uint32_t mage::DDS_HEADER::caps2
```

5.9.1.3 caps3

```
uint32_t mage::DDS_HEADER::caps3
```

5.9.1.4 caps4

```
uint32_t mage::DDS_HEADER::caps4
```

5.9.1.5 ddspf

```
DDS_PIXELFORMAT mage::DDS_HEADER::ddspf
```

5.9.1.6 depth

```
uint32_t mage::DDS_HEADER::depth
```

5.9.1.7 flags

```
uint32_t mage::DDS_HEADER::flags
```

5.9.1.8 height

```
uint32_t mage::DDS_HEADER::height
```

5.9.1.9 mip_map_count

```
uint32_t mage::DDS_HEADER::mip_map_count
```

5.9.1.10 pitch_or_linear_size

```
uint32_t mage::DDS_HEADER::pitch_or_linear_size
```

5.9.1.11 reserved1

```
uint32_t mage::DDS_HEADER::reserved1[11]
```

5.9.1.12 reserved2

```
uint32_t mage::DDS_HEADER::reserved2
```

5.9.1.13 size

```
uint32_t mage::DDS_HEADER::size
```

5.9.1.14 width

```
uint32_t mage::DDS_HEADER::width
```

5.10 mage::DDS_HEADER_DXT10 Struct Reference

Public Attributes

- DXGI_FORMAT dxgi_format
- uint32_t resource_dimension
- uint32_t misc_flag
- uint32_t array_size
- uint32_t misc_flags2

5.10.1 Member Data Documentation

5.10.1.1 array_size

```
uint32_t mage::DDS_HEADER_DXT10::array_size
```

5.10.1.2 dxgi_format

```
DXGI_FORMAT mage::DDS_HEADER_DXT10::dxgi_format
```

5.10.1.3 misc_flag

```
uint32_t mage::DDS_HEADER_DXT10::misc_flag
```

5.10.1.4 misc_flags2

```
uint32_t mage::DDS_HEADER_DXT10::misc_flags2
```

5.10.1.5 resource_dimension

```
uint32_t mage::DDS_HEADER_DXT10::resource_dimension
```

5.11 mage::DDS_PIXELFORMAT Struct Reference

Public Attributes

- uint32_t size
- uint32_t flags
- uint32_t fourCC
- uint32_t RGBBitCount
- uint32_t RBitMask
- uint32_t GBitMask
- uint32_t BBitMask
- uint32_t ABitMask

5.11.1 Member Data Documentation

5.11.1.1 ABitMask

```
uint32_t mage::DDS_PIXELFORMAT::ABitMask
```

5.11.1.2 BBitMask

```
uint32_t mage::DDS_PIXELFORMAT::BBitMask
```

5.11.1.3 flags

```
uint32_t mage::DDS_PIXELFORMAT::flags
```

5.11.1.4 fourCC

```
uint32_t mage::DDS_PIXELFORMAT::fourCC
```

5.11.1.5 GBitMask

```
uint32_t mage::DDS_PIXELFORMAT::GBitMask
```

5.11.1.6 RBitMask

```
uint32_t mage::DDS_PIXELFORMAT::RBitMask
```

5.11.1.7 RGBBitCount

```
uint32_t mage::DDS_PIXELFORMAT::RGBBitCount
```

5.11.1.8 size

```
uint32_t mage::DDS_PIXELFORMAT::size
```

5.12 mage::DeviceEnumeration Class Reference

```
#include <device_enumeration.hpp>
```

Public Member Functions

- IDXGIAdapter2 ∗ GetAdapter () const
- IDXGIOutput2 ∗ GetOutput () const
- const DXGI_MODE_DESC1 ∗ GetDisplayMode () const
- bool IsWindowed () const
- bool IsFullScreen () const
- bool IsVSynced () const

Protected Member Functions

- DeviceEnumeration ()
- virtual ∼DeviceEnumeration ()
- HRESULT InitializeAdapterAndOutput ()
- HRESULT UninitializeAdapterAndOutput ()
- HRESULT InitializeDisplayModes ()
- HRESULT Enumerate ()
- INT_PTR SettingsDialogProc (HWND hwndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)

Protected Attributes

- IDXGIAdapter2 ∗ m_adapter
- IDXGIOutput2 ∗ m_output
- VariableScript ∗ m_settings_script
- list< DXGI_MODE_DESC1 > m_display_modes
- const DXGI_MODE_DESC1 ∗ m_selected_diplay_mode
- bool m_windowed
- bool m_vsync

Friends

- class Engine
- INT_PTR CALLBACK SettingsDialogProcDelegate (HWND hwndDlg, UINT uMsg, WPARAM wParam, LP↩ ARAM lParam)

5.12.1 Detailed Description

A device enumeration.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 DeviceEnumeration()

```
mage::DeviceEnumeration::DeviceEnumeration ( )  [protected]
```

Constructs a device enumeration.

5.12.2.2 ∼DeviceEnumeration()

```
mage::DeviceEnumeration::∼DeviceEnumeration ( )  [protected], [virtual]
```

Destructs this device enumeration.

5.12.3 Member Function Documentation

5.12.3.1 Enumerate()

```
HRESULT mage::DeviceEnumeration::Enumerate ( )  [protected]
```

Enumerates the available display modes on the adapter output of the physical adapter with the most dedicated video memory.

Returns

A success/error value.

5.12.3.2 GetAdapter()

```
IDXGIAdapter2* mage::DeviceEnumeration::GetAdapter ( ) const
```

Returns the adapter.

Returns

A pointer to the adapter.

5.12.3.3 GetDisplayMode()

`const DXGI_MODE_DESC1* mage::DeviceEnumeration::GetDisplayMode ( ) const`

Returns the selected display mode by the user.

Returns

A pointer to the selected display mode.

5.12.3.4 GetOutput()

`IDXGIOutput2* mage::DeviceEnumeration::GetOutput ( ) const`

Returns the output.

Returns

A pointer to the output.

5.12.3.5 InitializeAdapterAndOutput()

`HRESULT mage::DeviceEnumeration::InitializeAdapterAndOutput ( ) [protected]`

Initializes the adapter and the output of this device enumeration.

Returns

A success/error value.

5.12.3.6 InitializeDisplayModes()

`HRESULT mage::DeviceEnumeration::InitializeDisplayModes ( ) [protected]`

Initializes the display modes of this device enumeration.

Returns

A success/error value.

5.12.3.7 IsFullScreen()

`bool mage::DeviceEnumeration::IsFullScreen ( ) const`

Checks whether the application should run in full screen mode.

Returns

`true` if the application should run in full screen mode. `false` otherwise.

5.12.3.8 IsVSynced()

```
bool mage::DeviceEnumeration::IsVSynced ( ) const
```

Checks whether v-sync should be enabled.

Returns

`true` if v-sync should be enabled. `false` otherwise.

5.12.3.9 IsWindowed()

```
bool mage::DeviceEnumeration::IsWindowed ( ) const
```

Checks whether the application should run in windowed mode.

Returns

`true` if the application should run in windowed mode. `false` otherwise.

5.12.3.10 SettingsDialogProc()

```
INT_PTR mage::DeviceEnumeration::SettingsDialogProc (
             HWND hwndDlg,
             UINT uMsg,
             WPARAM wParam,
             LPARAM lParam )  [protected]
```

Engine-defined callback function used with the CreateDialog for device enumeration.

Parameters

in	*hwndDlg*	A handle to the dialog box.
in	*uMsg*	The message.
in	*wParam*	Additional message-specific information.
in	*lParam*	Additional message-specific information.

Returns

`true` if *uMsg* is processed. `false` otherwise.

5.12.3.11 UninitializeAdapterAndOutput()

```
HRESULT mage::DeviceEnumeration::UninitializeAdapterAndOutput ( )  [protected]
```

Uninitializes the adapter and the output of this device enumeration.

Returns

A success/error value.

5.12.4 Friends And Related Function Documentation

5.12.4.1 Engine

```
friend class Engine  [friend]
```

5.12.4.2 SettingsDialogProcDelegate

```
INT_PTR CALLBACK SettingsDialogProcDelegate (
             HWND hwndDlg,
             UINT uMsg,
             WPARAM wParam,
             LPARAM lParam )  [friend]
```

Engine-defined callback function used with the CreateDialog for device enumeration.

Parameters

in	*hwndDlg*	A handle to the dialog box.
in	*uMsg*	The message.
in	*wParam*	Additional message-specific information.
in	*lParam*	Additional message-specific information.

Returns

`true` if *uMsg* is processed. `false` otherwise.

5.12.5 Member Data Documentation

5.12.5.1 m_adapter

```
IDXGIAdapter2* mage::DeviceEnumeration::m_adapter  [protected]
```

A pointer to the adapter (or video card).

5.12.5.2 m_display_modes

```
list< DXGI_MODE_DESC1 > mage::DeviceEnumeration::m_display_modes  [protected]
```

The linked list of enumerated display modes.

5.12.5.3 m_output

```
IDXGIOutput2* mage::DeviceEnumeration::m_output  [protected]
```

A pointer to the output.

5.12.5.4 m_selected_diplay_mode

```
const DXGI_MODE_DESC1∗ mage::DeviceEnumeration::m_selected_diplay_mode  [protected]
```

A pointer to the selected display mode by the user.

5.12.5.5 m_settings_script

```
VariableScript∗ mage::DeviceEnumeration::m_settings_script  [protected]
```

A pointer to the script which stores the device configuration.

5.12.5.6 m_vsync

```
bool mage::DeviceEnumeration::m_vsync  [protected]
```

Flag indicating whether v-sync should be enabled.

5.12.5.7 m_windowed

```
bool mage::DeviceEnumeration::m_windowed  [protected]
```

Flag indicating whether the application should run in windowed mode.

5.13 mage::Edge Struct Reference

```
#include <vertex.hpp>
```

Public Member Functions

- Edge (Vertex ∗v0, Vertex ∗v1)

Public Attributes

- Vertex ∗ v0
- Vertex ∗ v1

5.13.1 Detailed Description

A struct of edges.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 Edge()

```
mage::Edge::Edge (
            Vertex ∗ v0,
            Vertex ∗ v1 )
```

Constructs an edge between the two given vertices.

Parameters

in	*v0*	A pointer to the first vertex.
in	*v1*	A pointer to the second vertex.

5.13.3 Member Data Documentation

5.13.3.1 v0

```
Vertex* mage::Edge::v0
```

The first vertex of this edge.

5.13.3.2 v1

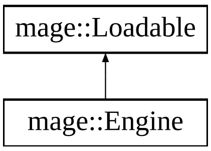
```
Vertex* mage::Edge::v1
```

The second vertex of this edge.

5.14 mage::Engine Class Reference

```
#include <engine.hpp>
```

Inheritance diagram for mage::Engine:



Public Member Functions

- Engine (const EngineSetup ∗setup=nullptr)
- virtual ∼Engine ()
- void Run (int nCmdShow=SW_NORMAL)
- const MainWindow ∗ GetMainWindow () const
- void SetDeactiveFlag (bool deactive)
- Renderer ∗ GetRenderer () const
- void SetModeSwitchFlag (bool mode_switch)
- StateManager ∗ GetStateManager () const
- ResourceManager< VariableScript > ∗ GetScriptManager () const
- const InputManager ∗ GetInputManager () const

Protected Member Functions

- HRESULT InitializeSystems (const EngineSetup ∗setup)
- HRESULT UninitializeSystems ()

Protected Attributes

- MainWindow ∗ m_main_window
- bool m_deactive
- Renderer ∗ m_renderer
- bool m_mode_switch
- StateManager ∗ m_state_manager
- ResourceManager< VariableScript > ∗ m_script_manager
- InputManager ∗ m_input_manager

5.14.1 Detailed Description

A class of engines.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 Engine()

```
mage::Engine::Engine (
            const EngineSetup ∗ setup = nullptr )
```

Constructs an engine from the given engine setup.

Parameters

in	*setup*	A pointer to an engine setup.

5.14.2.2 ∼Engine()

```
mage::Engine::∼Engine ( ) [virtual]
```

Destructs this engine.

5.14.3 Member Function Documentation

5.14.3.1 GetInputManager()

```
const InputManager∗ mage::Engine::GetInputManager ( ) const
```

Returns the input manager of this engine.

Returns

A pointer to the input manager of this engine.

5.14.3.2 GetMainWindow()

```
const MainWindow* mage::Engine::GetMainWindow ( ) const
```

Returns the main window of this engine.

Returns

The main window of this engine.

5.14.3.3 GetRenderer()

```
Renderer* mage::Engine::GetRenderer ( ) const
```

Returns the renderer of this engine.

Returns

A pointer to the renderer of this engine.

5.14.3.4 GetScriptManager()

```
ResourceManager< VariableScript >* mage::Engine::GetScriptManager ( ) const
```

Returns the script manager of this engine.

Returns

A pointer to the script manager of this engine.

5.14.3.5 GetStateManager()

```
StateManager* mage::Engine::GetStateManager ( ) const
```

Returns the state manager of this engine.

Returns

A pointer to the state manager of this engine.

5.14.3.6 InitializeSystems()

```
HRESULT mage::Engine::InitializeSystems (
             const EngineSetup * setup )  [protected]
```

Initializes the different systems of this engine.

Parameters

in	*setup*	A pointer to an engine setup.

Returns

A success/error value.

5.14.3.7 Run()

```
void mage::Engine::Run (
            int nCmdShow = SW_NORMAL )
```

Runs this engine.

Parameters

in	*nCmdShow*	Controls how the engine window is to be shown.

5.14.3.8 SetDeactiveFlag()

```
void mage::Engine::SetDeactiveFlag (
            bool deactive )
```

Sets the deactive flag of this engine to the given value.

Parameters

in	*deactive*	The new value for the deactive flag.

5.14.3.9 SetModeSwitchFlag()

```
void mage::Engine::SetModeSwitchFlag (
            bool mode_switch )
```

Sets the mode switch flag of this engine to the given value.

Parameters

in	*mode_switch*	The new value for the mode switch flag.

5.14.3.10 UninitializeSystems()

```
HRESULT mage::Engine::UninitializeSystems ( )  [protected]
```

Unitialize the different systems of this engine.

Returns

A success/error value.

5.14.4 Member Data Documentation

5.14.4.1 m_deactive

```
bool mage::Engine::m_deactive  [protected]
```

Flag indicating whether the application is active or not.

5.14.4.2 m_input_manager

```
InputManager* mage::Engine::m_input_manager  [protected]
```

A pointer to the input manager of this engine.

5.14.4.3 m_main_window

```
MainWindow* mage::Engine::m_main_window  [protected]
```

A pointer to the main window of this engine.

5.14.4.4 m_mode_switch

```
bool mage::Engine::m_mode_switch  [protected]
```

Flag indicating whether the application should switch between full screen and windowed mode.

5.14.4.5 m_renderer

```
Renderer* mage::Engine::m_renderer  [protected]
```

A pointer to the renderer of this engine.

5.14.4.6 m_script_manager

```
ResourceManager< VariableScript >* mage::Engine::m_script_manager  [protected]
```

A pointer the script manager of this engine

5.14.4.7 m_state_manager

```
StateManager* mage::Engine::m_state_manager  [protected]
```

A pointer to the state manager of this engine.

5.15 mage::EngineSetup Struct Reference

```
#include <engine.hpp>
```

Public Member Functions

- EngineSetup (const wstring &name=L"Application")
- EngineSetup (const EngineSetup ∗setup)

Public Attributes

- HINSTANCE m_hinstance
- wstring m_name
- void(∗ StateSetup)()

5.15.1 Detailed Description

A struct of engine setups.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 EngineSetup() [1/2]

```
mage::EngineSetup::EngineSetup (
             const wstring & name = L"Application" )
```

Constructs an engine setup with the given application name.

Parameters

in	*name*	A reference to the name of the application.

5.15.2.2 EngineSetup() [2/2]

```
mage::EngineSetup::EngineSetup (
             const EngineSetup * setup )
```

Constructs an engine setup from the given engine setup.

Precondition

setup does not point to `nullptr`.

Parameters

in	*setup*	A pointer to the engine setup.

5.15.3 Member Data Documentation

5.15.3.1 m_hinstance

```
HINSTANCE mage::EngineSetup::m_hinstance
```

Application instance handle.

5.15.3.2 m_name

```
wstring mage::EngineSetup::m_name
```

Name of the application.

5.15.3.3 StateSetup

```
void(* mage::EngineSetup::StateSetup) ()
```

The state setup function.

5.16 mage::Face Struct Reference

```
#include <vertex.hpp>
```

Public Member Functions

- Face (Vertex ∗v0, Vertex ∗v1, Vertex ∗v2)

Public Attributes

- Vertex ∗ v0
- Vertex ∗ v1
- Vertex ∗ v2

5.16.1 Detailed Description

A struct of faces.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Face()

```
mage::Face::Face (
             Vertex ∗ v0,
             Vertex ∗ v1,
             Vertex ∗ v2 )
```

Constructs a face for the three given vertices.

Parameters

in	*v0*	A pointer to the first vertex.
in	*v1*	A pointer to the second vertex.
in	*v2*	A pointer to the third vertex.

5.16.3 Member Data Documentation

5.16.3.1 v0

```
Vertex* mage::Face::v0
```

The first vertex of this face.

5.16.3.2 v1

```
Vertex* mage::Face::v1
```

The second vertex of this face.

5.16.3.3 v2

```
Vertex* mage::Face::v2
```

The third vertex of this face.

5.17 mage::Font Class Reference

```
#include <font.hpp>
```

Public Member Functions

- Font (const wstring &name=L"Arial", uint16_t size=10, uint32_t bold=FW_NORMAL, bool italic=false)
- virtual ∼Font ()
- void Render (char ∗text, float x, float y, XMFLOAT4 colour=XMFLOAT4(1.0f, 1.0f, 1.0f, 1.0f))

Protected Member Functions

- bool PrepareFont (HDC hDC, bool measure=false)

Protected Attributes

- ID3D11Buffer * m_vertex_buffer
- ID3D11Texture2D * m_texture
- uint32_t m_texture_width
- uint32_t m_texture_height
- float m_texture_coords [96][4]
- uint16_t m_spacing

5.17.1 Constructor & Destructor Documentation

5.17.1.1 Font()

```
mage::Font::Font (
            const wstring & name = L"Arial",
            uint16_t size = 10,
            uint32_t bold = FW_NORMAL,
            bool italic = false )
```

5.17.1.2 ∼Font()

```
mage::Font::∼Font ( )  [virtual]
```

5.17.2 Member Function Documentation

5.17.2.1 PrepareFont()

```
bool mage::Font::PrepareFont (
            HDC hDC,
            bool measure = false )  [protected]
```

5.17.2.2 Render()

```
void mage::Font::Render (
            char * text,
            float x,
            float y,
            XMFLOAT4 colour = XMFLOAT4(1.0f, 1.0f, 1.0f, 1.0f) )
```

5.17.3 Member Data Documentation

5.17.3.1 m_spacing

```
uint16_t mage::Font::m_spacing  [protected]
```

5.17.3.2 m_texture

```
ID3D11Texture2D* mage::Font::m_texture  [protected]
```

5.17.3.3 m_texture_coords

```
float mage::Font::m_texture_coords[96][4]  [protected]
```

5.17.3.4 m_texture_height

```
uint32_t mage::Font::m_texture_height  [protected]
```

5.17.3.5 m_texture_width

```
uint32_t mage::Font::m_texture_width  [protected]
```

5.17.3.6 m_vertex_buffer

```
ID3D11Buffer* mage::Font::m_vertex_buffer  [protected]
```

5.18 mage::IndexedEdge Struct Reference

```
#include <vertex.hpp>
```

Public Attributes

- uint32_t iv0
- uint32_t iv1

5.18.1 Detailed Description

A struct of indexed edges.

5.18.2 Member Data Documentation

5.18.2.1 iv0

```
uint32_t mage::IndexedEdge::iv0
```

The index of the edge's first vertex.

5.18.2.2 iv1

```
uint32_t mage::IndexedEdge::iv1
```

The index of the edge's second vertex.

5.19 mage::IndexedFace Struct Reference

```
#include <vertex.hpp>
```

Public Attributes

- uint32_t iv0
- uint32_t iv1
- uint32_t iv2

5.19.1 Detailed Description

A struct of indexed faces.

5.19.2 Member Data Documentation

5.19.2.1 iv0

```
uint32_t mage::IndexedFace::iv0
```

Index of the face's first vertex.

5.19.2.2 iv1

```
uint32_t mage::IndexedFace::iv1
```

Index of the face's second vertex.

5.19.2.3 iv2

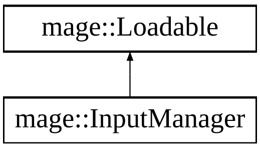
```
uint32_t mage::IndexedFace::iv2
```

Index of the face's third vertex.

5.20 mage::InputManager Class Reference

```
#include <input_manager.hpp>
```

Inheritance diagram for mage::InputManager:



Public Member Functions

- const Keyboard ∗ GetKeyboard () const
- const Mouse ∗ GetMouse () const

Protected Member Functions

- InputManager (HWND hwindow)
- virtual ∼InputManager ()
- HRESULT InitializeDI ()
- HRESULT UninitializeDI ()
- HRESULT InitializeInputSystems ()
- HRESULT UninitializeInputSystems ()
- void Update ()

Protected Attributes

- HWND m_hwindow
- IDirectInput8 ∗ m_di
- Keyboard ∗ m_keyboard
- Mouse ∗ m_mouse

Friends

- class Engine

5.20.1 Detailed Description

A class of input managers.

5.20.2 Constructor & Destructor Documentation

5.20.2.1 InputManager()

```
mage::InputManager::InputManager (
            HWND hwindow ) [protected]
```

Constructs an input manager for the given window handle.

Parameters

in	*hwindow*	The handle of the parent window.

5.20.2.2 ∼InputManager()

```
mage::InputManager::∼InputManager ( )  [protected], [virtual]
```

Destructs this input manager.

5.20.3 Member Function Documentation

5.20.3.1 GetKeyboard()

```
const Keyboard* mage::InputManager::GetKeyboard ( ) const
```

Returns the keyboard of this input manager.

Returns

A pointer to the keyboard of this input manager.

5.20.3.2 GetMouse()

```
const Mouse* mage::InputManager::GetMouse ( ) const
```

Returns the mouse of this input manager.

Returns

A pointer to the mouse of this input manager.

5.20.3.3 InitializeDI()

```
HRESULT mage::InputManager::InitializeDI ( )  [protected]
```

Initializes the DirectInput object of this input manager.

Returns

A success/error value.

5.20.3.4 InitializeInputSystems()

```
HRESULT mage::InputManager::InitializeInputSystems ( )  [protected]
```

Initializes the different input systems of this input manager.

5.20.3.5 UninitializeDI()

```
HRESULT mage::InputManager::UninitializeDI ( )  [protected]
```

Uninitializes the DirectInput object of this input manager.

Returns

A success/error value.

5.20.3.6 UninitializeInputSystems()

```
HRESULT mage::InputManager::UninitializeInputSystems ( )  [protected]
```

Initializes the different input systems of this manager.

5.20.3.7 Update()

```
void mage::InputManager::Update ( )  [protected]
```

Updates the state of the input systems of this input manager.

5.20.4 Friends And Related Function Documentation

5.20.4.1 Engine

```
friend class Engine  [friend]
```

5.20.5 Member Data Documentation

5.20.5.1 m_di

```
IDirectInput8* mage::InputManager::m_di  [protected]
```

The DirectInput object of this input manager.

The methods of the IDirectInput8 interface are used to enumerate, create, and retrieve the status of Microsoft DirectInput device.

5.20.5.2 m_hwindow

```
HWND mage::InputManager::m_hwindow  [protected]
```

The handle of the parent window.

5.20.5.3 m_keyboard

```
Keyboard* mage::InputManager::m_keyboard [protected]
```

A pointer to the keyboard of this input manager.

5.20.5.4 m_mouse

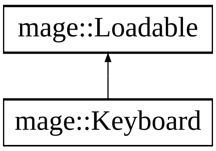
```
Mouse* mage::InputManager::m_mouse [protected]
```

A pointer to the mouse of this input manager.

5.21 mage::Keyboard Class Reference

```
#include <keyboard.hpp>
```

Inheritance diagram for mage::Keyboard:



Public Member Functions

- bool GetKeyPress (char key, bool ignore_press_stamp=false) const

Protected Member Functions

- Keyboard (HWND hwindow, IDirectInput8 ∗di)
- virtual ∼Keyboard ()
- HRESULT InitializeKeyboard (IDirectInput8 ∗di)
- HRESULT UninitializeKeyboard ()
- void Update ()

Protected Attributes

- HWND m_hwindow
- uint64_t m_press_stamp
- IDirectInputDevice8 ∗ m_keyboard
- char m_key_state [256]
- uint64_t m_key_press_stamp [256]

Friends

- class InputManager

5.21.1 Detailed Description

A class of keyboards.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 Keyboard()

```
mage::Keyboard::Keyboard (
            HWND hwindow,
            IDirectInput8 * di )  [protected]
```

Constructs a keyboard.

Parameters

in	*hwindow*	The handle of the parent window.
in	*di*	A pointer to a direct input object.

5.21.2.2 ∼Keyboard()

```
mage::Keyboard::∼Keyboard ( )  [protected], [virtual]
```

Destructs this keyboard.

5.21.3 Member Function Documentation

5.21.3.1 GetKeyPress()

```
bool mage::Keyboard::GetKeyPress (
            char key,
            bool ignore_press_stamp = false ) const
```

Checks whether the given key of this keyboard is pressed.

Parameters

in	*key*	The key.
in	*ignore_press_stamp*	Flag indicating whether press stamps should be ignored. Consistent presses will return false when using the press stamp.

Returns

`true` if the given key of this keyboard is pressed. `false` otherwise.

5.21.3.2 InitializeKeyboard()

```
HRESULT mage::Keyboard::InitializeKeyboard (
             IDirectInput8 * di )  [protected]
```

Initializes the keyboard device of this keyboard.

Parameters

in	*di*	A pointer to a direct input object.

Returns

A success/error value.

5.21.3.3 UninitializeKeyboard()

```
HRESULT mage::Keyboard::UninitializeKeyboard ( )  [protected]
```

Uninitializes the keyboard device of this keyboard.

Returns

A success/error value.

5.21.3.4 Update()

```
void mage::Keyboard::Update ( )  [protected]
```

Updates the state of this keyboard.

5.21.4 Friends And Related Function Documentation

5.21.4.1 InputManager

```
friend class InputManager  [friend]
```

5.21.5 Member Data Documentation

5.21.5.1 m_hwindow

```
HWND mage::Keyboard::m_hwindow  [protected]
```

The handle of the parent window.

5.21.5.2 m_key_press_stamp

```
uint64_t mage::Keyboard::m_key_press_stamp[256]  [mutable], [protected]
```

Stamps the keys pressed in the last frame of this keyboard.

5.21.5.3 m_key_state

```
char mage::Keyboard::m_key_state[256]  [protected]
```

State of the keys of this keyboard.

5.21.5.4 m_keyboard

```
IDirectInputDevice8* mage::Keyboard::m_keyboard  [protected]
```

The DirectInput keyboard device of this keyboard.

The methods of the IDirectInputDevice8 interface are used to gain and release access to Microsoft DirectInput devices, manage device properties and information, set behavior, perform initialization, create and play force-feedback effects, and invoke a device's control panel.

5.21.5.5 m_press_stamp

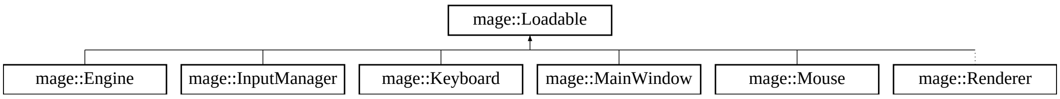
```
uint64_t mage::Keyboard::m_press_stamp  [protected]
```

The current press stamp (incremented every frame).

5.22 mage::Loadable Class Reference

```
#include <loadable.hpp>
```

Inheritance diagram for mage::Loadable:



Public Member Functions

- bool IsLoaded () const

Protected Member Functions

- Loadable (bool loaded=false)
- virtual ∼Loadable ()
- void SetLoaded (bool loaded=true)

Private Attributes

- bool m_loaded

5.22.1 Detailed Description

A class of loadables.

5.22.2 Constructor & Destructor Documentation

5.22.2.1 Loadable()

```
mage::Loadable::Loadable (
             bool loaded = false )  [protected]
```

Constructs a loadable.

Parameters

in	*loaded*	Flag indicating wether the loadable is loaded.

5.22.2.2 ∼Loadable()

```
virtual mage::Loadable::∼Loadable ( )  [protected], [virtual]
```

Destructs this loadable.

5.22.3 Member Function Documentation

5.22.3.1 IsLoaded()

```
bool mage::Loadable::IsLoaded ( ) const
```

Checks wether this loadable is loaded.

Returns

`true` if this loadable is loaded. `false` otherwise.

5.22.3.2 SetLoaded()

```
void mage::Loadable::SetLoaded (
             bool loaded = true )  [protected]
```

Set the state of this loadable to the given value.

Parameters

`in`	*loaded*	Flag indicating wether this loadable is loaded.

5.22.4 Member Data Documentation

5.22.4.1 m_loaded

```
bool mage::Loadable::m_loaded  [private]
```

Flag indicating wether this loadable is loaded.

5.23 mage::LoggingConfiguration Struct Reference

```
#include <logging.hpp>
```

Public Member Functions

- LoggingConfiguration ()
- bool IsQuiet () const
- bool IsVerbose () const

Private Attributes

- bool m_quiet
- bool m_verbose

5.23.1 Detailed Description

A struct of logging configurations of the engine processing.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 LoggingConfiguration()

```
mage::LoggingConfiguration::LoggingConfiguration ( )
```

Constructs a new logging configuration.

5.23.3 Member Function Documentation

5.23.3.1 IsQuiet()

```
bool mage::LoggingConfiguration::IsQuiet ( ) const
```

Checks whether the logging of the engine processing is quiet.

Returns

`true` if the logging of the engine processing is quiet. `false` otherwise.

5.23.3.2 IsVerbose()

```
bool mage::LoggingConfiguration::IsVerbose ( ) const
```

Checks wheter the logging of the engine processing is verbose.

Returns

`true` if the logging of the engine processing is verbose. `false` otherwise.

5.23.4 Member Data Documentation

5.23.4.1 m_quiet

```
bool mage::LoggingConfiguration::m_quiet  [private]
```

Flag indicating the logging of the engine processing is quiet.

5.23.4.2 m_verbose

```
bool mage::LoggingConfiguration::m_verbose  [private]
```

Flag indicating the logging of the engine processing is verbose.

5.24 mage::LVertex Struct Reference

```
#include <vertex.hpp>
```

Public Member Functions

- LVertex ()
- LVertex (XMFLOAT3 p, XMFLOAT4 diffuse, XMFLOAT2 tex)

Public Attributes

- XMFLOAT3 p
- XMFLOAT4 diffuse
- XMFLOAT2 tex

5.24.1 Detailed Description

A struct of lit vertices.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 LVertex() [1/2]

```
mage::LVertex::LVertex ( )
```

Constructs a lit vertex.

5.24.2.2 LVertex() [2/2]

```
mage::LVertex::LVertex (
             XMFLOAT3 p,
             XMFLOAT4 diffuse,
             XMFLOAT2 tex )
```

Constructs a lit vertex.

Parameters

in	*p*	The position of the lit vertex (in object space).
in	*diffuse*	The diffuse colour of the lit vertex.
in	*tex*	The texture coordinate of the lit vertex.

5.24.3 Member Data Documentation

5.24.3.1 diffuse

```
XMFLOAT4 mage::LVertex::diffuse
```

The diffuse colour of this lit vertex.

5.24.3.2 p

```
XMFLOAT3 mage::LVertex::p
```

The position of this lit vertex (in object space).

5.24.3.3 tex

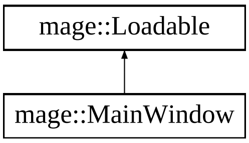
```
XMFLOAT2 mage::LVertex::tex
```

The texture coordinates of this lit vertex.

5.25 mage::MainWindow Class Reference

```
#include <main_window.hpp>
```

Inheritance diagram for mage::MainWindow:



Public Member Functions

- HWND GetHandle () const
- const wstring & GetName () const

Protected Member Functions

- MainWindow (HINSTANCE hinstance, wstring name, LONG width, LONG height)
- virtual ∼MainWindow ()
- HRESULT InitializeWindow (LONG width, LONG height)
- HRESULT InitializeWindow (RECT rectangle)
- HRESULT UninitializeWindow ()
- BOOL Show (int nCmdShow)

Protected Attributes

- HINSTANCE m_hinstance
- HWND m_hwindow
- wstring m_name

Friends

- class Engine

5.25.1 Detailed Description

A class of main windows.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 MainWindow()

```
mage::MainWindow::MainWindow (
             HINSTANCE hinstance,
             wstring name,
             LONG width,
             LONG height )  [protected]
```

Constructs a main window.

Parameters

in	*hinstance*	The application instance handle.
in	*name*	The application name.
in	*width*	The width of the window.
in	*height*	The height of the window.

5.25.2.2 ∼MainWindow()

```
mage::MainWindow::∼MainWindow ( )  [protected], [virtual]
```

Destructs this main window.

5.25.3 Member Function Documentation

5.25.3.1 GetHandle()

```
HWND mage::MainWindow::GetHandle ( ) const
```

Returns the window handle of this main window.

Returns

The window handle of this main window.

5.25.3.2 GetName()

```
const wstring& mage::MainWindow::GetName ( ) const
```

Returns the name of this main window.

Returns

The name of this main window.

5.25.3.3 InitializeWindow() [1/2]

```
HRESULT mage::MainWindow::InitializeWindow (
             LONG width,
             LONG height )  [protected]
```

Initializes the engine window of this main window.

Parameters

in	*width*	The width of the client rectangle of the window.
in	*height*	The height of the client rectangle of the window.

Returns

A success/error value.

5.25.3.4 InitializeWindow() [2/2]

```
HRESULT mage::MainWindow::InitializeWindow (
             RECT rectangle )  [protected]
```

Initializes the engine window of this main window.

Parameters

in	*rectangle*	The client rectangle of the window.

Returns

A success/error value.

5.25.3.5 Show()

```
BOOL mage::MainWindow::Show (
             int nCmdShow )  [protected]
```

Sets the specified window's show state of this main window.

Parameters

in	*nCmdShow*	Controls how this window is to be shown.

Returns

If the window was previously visible, the return value is nonzero. If the window was previously hidden, the return value is zero.

5.25.3.6 UninitializeWindow()

```
HRESULT mage::MainWindow::UninitializeWindow ( )  [protected]
```

Unitializes the engine window of this main window.

Returns

A success/error value.

5.25.4 Friends And Related Function Documentation

5.25.4.1 Engine

```
friend class Engine  [friend]
```

5.25.5 Member Data Documentation

5.25.5.1 m_hinstance

```
HINSTANCE mage::MainWindow::m_hinstance  [protected]
```

Application instance handle.

5.25.5.2 m_hwindow

```
HWND mage::MainWindow::m_hwindow  [protected]
```

Window handle of this main window.

5.25.5.3 m_name

```
wstring mage::MainWindow::m_name  [protected]
```

The name of this main window.

5.26 mage::MemoryArena Class Reference

```
#include <arena.hpp>
```

Public Member Functions

- MemoryArena (uint32_t block_size=32768)
- ∼MemoryArena ()
- size_t GetBlockSize () const
- size_t GetCurrentBlockSize () const
- size_t GetTotalBlockSize () const
- char ∗ GetCurrentBlockPtr () const
- void Reset ()
- void ∗ Alloc (size_t size)
- template<typename T >
  T ∗ Alloc (size_t count=1, bool initialization=true)

Private Member Functions

- MemoryArena (const MemoryArena &arena)=delete
- MemoryArena & operator= (const MemoryArena &arena)=delete

Private Attributes

- const size_t m_block_size
- size_t m_current_block_pos
- pair< size_t, char ∗> m_current_block
- list< pair< size_t, char ∗> > m_used_blocks
- list< pair< size_t, char ∗> > m_available_blocks

5.26.1 Detailed Description

A class of memory arena's.

5.26.2 Constructor & Destructor Documentation

5.26.2.1 MemoryArena() [1/2]

```
mage::MemoryArena::MemoryArena (
            uint32_t block_size = 32768 )
```

Constructs a memory arena with given block size.

Parameters

in	*block_size*	The maximum block size in bytes.

5.26.2.2 ∼MemoryArena()

```
mage::MemoryArena::∼MemoryArena ( )
```

Destructs the given memory arena.

5.26.2.3 MemoryArena() [2/2]

```
mage::MemoryArena::MemoryArena (
            const MemoryArena & arena )  [private], [delete]
```

Constructs a memory arena from the given memory arena.

Parameters

in	*arena*	The memory arena.

5.26.3 Member Function Documentation

5.26.3.1 Alloc() [1/2]

```
void* mage::MemoryArena::Alloc (
            size_t size )
```

Allocates a block of memory of the given size.

Parameters

in	*size*	The requested size in bytes to allocate in memory.

Returns

`nullptr` if the allocation failed.
A pointer to the memory block that was allocated.

5.26.3.2 Alloc() [2/2]

```
template<typename T >
T* mage::MemoryArena::Alloc (
             size_t count = 1,
             bool initialization = true )
```

Allocates a block of memory.

Template Parameters

T	The type of objects to allocate in memory.

Parameters

in	*count*	The number of objects of type `T` to allocate in memory.
in	*initialization*	Flag indicating whether the objects need to be initialized (i.e. the constructor needs to be called).

Returns

`nullptr` if the allocation failed.
A pointer to the memory block that was allocated.

Note

The objects will be constructed with their default empty constructor.

5.26.3.3 GetBlockSize()

```
size_t mage::MemoryArena::GetBlockSize ( ) const
```

Returns the maximum block size of this memory arena.

Returns

The maximum block size of this memory arena.

5.26.3.4 GetCurrentBlockPtr()

```
char* mage::MemoryArena::GetCurrentBlockPtr ( ) const
```

Returns a pointer to the current block of this memory arena.

Returns

A pointer to the current block of this memory arena.

5.26.3.5 GetCurrentBlockSize()

```
size_t mage::MemoryArena::GetCurrentBlockSize ( ) const
```

Returns the block size (in bytes) of the current block of this memory arena.

Returns

The block size (in bytes) of the current block of this memory arena.

5.26.3.6 GetTotalBlockSize()

```
size_t mage::MemoryArena::GetTotalBlockSize ( ) const
```

Returns the block size (in bytes) of all blocks of this memory arena.

Returns

The block size (in bytes) of all blocks of this memory arena.

5.26.3.7 operator=()

```
MemoryArena& mage::MemoryArena::operator= (
             const MemoryArena & arena )  [private], [delete]
```

Copies the given memory arena to this memory arena.

Parameters

in	*arena*	The memory arena.

5.26.3.8 Reset()

```
void mage::MemoryArena::Reset ( )
```

Resets this memory arena.

5.26.4 Member Data Documentation

5.26.4.1 m_available_blocks

```
list< pair< size_t, char * > > mage::MemoryArena::m_available_blocks  [private]
```

Pointers to the available blocks of this memory arena.

5.26.4.2 m_block_size

```
const size_t mage::MemoryArena::m_block_size  [private]
```

The fixed block size of this memory arena.

5.26.4.3 m_current_block

```
pair< size_t, char * > mage::MemoryArena::m_current_block  [private]
```

A pointer to the current block of this memory arena.

5.26.4.4 m_current_block_pos

```
size_t mage::MemoryArena::m_current_block_pos  [private]
```

The current block position of this memory arena.

5.26.4.5 m_used_blocks

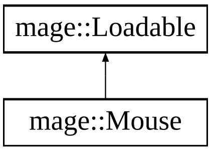
```
list< pair< size_t, char * > > mage::MemoryArena::m_used_blocks  [private]
```

Pointers to the used blocks of this memory arena.

5.27 mage::Mouse Class Reference

```
#include <mouse.hpp>
```

Inheritance diagram for mage::Mouse:



Public Member Functions

- bool GetMouseButtonPress (char mouse_button, bool ignore_press_stamp=false) const
- long GetPosX () const
- long GetPosY () const
- long GetDeltaX () const
- long GetDeltaY () const
- long GetDeltaWheel () const

Protected Member Functions

- Mouse (HWND hwindow, IDirectInput8 ∗di)
- virtual ∼Mouse ()
- HRESULT InitializeMouse (IDirectInput8 ∗di)
- HRESULT UninitializeMouse ()
- void Update ()

Protected Attributes

- HWND m_hwindow
- uint64_t m_press_stamp
- IDirectInputDevice8 ∗ m_mouse
- DIMOUSESTATE m_mouse_state
- uint64_t m_mouse_button_press_stamp [3]
- POINT m_mouse_position

Friends

- class InputManager

5.27.1 Detailed Description

A class of mouses.

5.27.2 Constructor & Destructor Documentation

5.27.2.1 Mouse()

```
mage::Mouse::Mouse (
            HWND hwindow,
            IDirectInput8 ∗ di )  [protected]
```

Constructs a mouse.

Parameters

in	*hwindow*	The handle of the parent window.
in	*di*	A pointer to a direct input object.

5.27.2.2 ∼Mouse()

```
mage::Mouse::∼Mouse ( )  [protected], [virtual]
```

Destructs this mouse.

5.27.3 Member Function Documentation

5.27.3.1 GetDeltaWheel()

```
long mage::Mouse::GetDeltaWheel ( ) const
```

Returns the change in this mouse's scroll wheel.

Returns

The change in this mouse's scroll wheel.

5.27.3.2 GetDeltaX()

```
long mage::Mouse::GetDeltaX ( ) const
```

Returns the change in this mouse's horizontal coordinate.

Returns

The change in this mouse's horizontal coordinate.

5.27.3.3 GetDeltaY()

```
long mage::Mouse::GetDeltaY ( ) const
```

Returns the change in this mouse's vertical coordinate.

Returns

The change in this mouse's vertical coordinate.

5.27.3.4 GetMouseButtonPress()

```
bool mage::Mouse::GetMouseButtonPress (
            char mouse_button,
            bool ignore_press_stamp = false ) const
```

Checks whether the given mouse button of this mouse is pressed.

Parameters

in	*mouse_button*	The mouse button.
in	*ignore_press_stamp*	Flag indicating whether press stamps should be ignored. Consistent presses will return false when using the press stamp.

Returns

`true` if the given mouse button is pressed. `false` otherwise.

5.27.3.5 GetPosX()

```
long mage::Mouse::GetPosX ( ) const
```

Returns the horizontal position of this mouse.

Returns

The horizontal position of this mouse.

5.27.3.6 GetPosY()

```
long mage::Mouse::GetPosY ( ) const
```

Returns the vertical position of this mouse.

Returns

The vertical position of this mouse.

5.27.3.7 InitializeMouse()

```
HRESULT mage::Mouse::InitializeMouse (
             IDirectInput8 * di ) [protected]
```

Initializes the mouse device of this mouse.

Parameters

in	*di*	A pointer to a direct input object.

Returns

A success/error value.

5.27.3.8 UninitializeMouse()

```
HRESULT mage::Mouse::UninitializeMouse ( ) [protected]
```

Uninitializes the mouse device of this mouse.

Returns

A success/error value.

5.27.3.9 Update()

```
void mage::Mouse::Update ( )  [protected]
```

Updates the state of this mouse.

5.27.4 Friends And Related Function Documentation

5.27.4.1 InputManager

```
friend class InputManager  [friend]
```

5.27.5 Member Data Documentation

5.27.5.1 m_hwindow

```
HWND mage::Mouse::m_hwindow  [protected]
```

The handle of the parent window.

5.27.5.2 m_mouse

```
IDirectInputDevice8* mage::Mouse::m_mouse  [protected]
```

DirectInput mouse device of this mouse.

The methods of the IDirectInputDevice8 interface are used to gain and release access to Microsoft DirectInput devices, manage device properties and information, set behavior, perform initialization, create and play force-feedback effects, and invoke a device's control panel.

5.27.5.3 m_mouse_button_press_stamp

```
uint64_t mage::Mouse::m_mouse_button_press_stamp[3]  [mutable], [protected]
```

Stamps the mouse buttons pressed in the last frame of this mouse.

5.27.5.4 m_mouse_position

```
POINT mage::Mouse::m_mouse_position  [protected]
```

The position of the mouse cursor on the screen of this mouse.

5.27.5.5 m_mouse_state

```
DIMOUSESTATE mage::Mouse::m_mouse_state  [protected]
```

State of the mouse buttons of this mouse.

Describes the state of a mouse device that has up to four buttons, or another device that is being accessed as if it were a mouse device.

5.27.5.6 m_press_stamp

```
uint64_t mage::Mouse::m_press_stamp  [protected]
```

The current press stamp (incremented every frame).

5.28 mage::Mutex Class Reference

```
#include <lock.hpp>
```

Static Public Member Functions

- static Mutex ∗ Create ()
- static void Destroy (Mutex ∗mutex)

Private Member Functions

- Mutex ()
- Mutex (Mutex &mutex)
- ∼Mutex ()
- Mutex & operator= (const Mutex &mutex)

Private Attributes

- CRITICAL_SECTION m_critical_section

Friends

- struct MutexLock

5.28.1 Detailed Description

A class of mutexes.

5.28.2 Constructor & Destructor Documentation

5.28.2.1 Mutex() [1/2]

```
mage::Mutex::Mutex ( )  [private]
```

Constructs a mutex.

5.28.2.2 Mutex() [2/2]

```
mage::Mutex::Mutex (
             Mutex & mutex )  [private]
```

Constructs a mutex from the given mutex.

Parameters

in	*mutex*	A reference to a mutex.

5.28.2.3 ∼Mutex()

```
mage::Mutex::∼Mutex ( )  [private]
```

Destructs this mutex.

5.28.3 Member Function Documentation

5.28.3.1 Create()

```
static Mutex∗ mage::Mutex::Create ( )  [static]
```

Creates a mutex.

5.28.3.2 Destroy()

```
static void mage::Mutex::Destroy (
             Mutex ∗ mutex )  [static]
```

Destroys a given mutex.

Parameters

in	*mutex*	The mutex to destroy.

5.28.3.3 operator=()

```
Mutex& mage::Mutex::operator= (
             const Mutex & mutex )  [private]
```

Copies the given mutex to this mutex.

Parameters

in	*mutex*	A reference to a mutex.

Returns

A reference to the copy of *mutex*.

5.28.4 Friends And Related Function Documentation

5.28.4.1 MutexLock

```
friend struct MutexLock  [friend]
```

5.28.5 Member Data Documentation

5.28.5.1 m_critical_section

```
CRITICAL_SECTION mage::Mutex::m_critical_section  [private]
```

The critical section object of this mutex.

5.29 mage::MutexLock Struct Reference

```
#include <lock.hpp>
```

Public Member Functions

- MutexLock (Mutex &mutex)
- ∼MutexLock ()

Private Member Functions

- MutexLock (const MutexLock &mutex_lock)
- MutexLock & operator= (const MutexLock &mutex_lock)

Private Attributes

- Mutex & m_mutex

5.29.1 Detailed Description

A struct of mutex locks.

5.29.2 Constructor & Destructor Documentation

5.29.2.1 MutexLock() [1/2]

```
mage::MutexLock::MutexLock (
            Mutex & mutex )
```

Constructs a mutex lock for the given mutex.

Parameters

in	*mutex*	A reference to a mutex.

5.29.2.2 ∼MutexLock()

```
mage::MutexLock::∼MutexLock ( )
```

Destructs this mutex lock.

5.29.2.3 MutexLock() [2/2]

```
mage::MutexLock::MutexLock (
             const MutexLock & mutex_lock )  [private]
```

Constructs a mutex lock from the given mutex lock.

Parameters

in	*mutex_lock*	A reference to a mutex lock.

5.29.3 Member Function Documentation

5.29.3.1 operator=()

```
MutexLock& mage::MutexLock::operator= (
             const MutexLock & mutex_lock )  [private]
```

Copies the given mutex lock to this mutex lock.

Parameters

in	*mutex_lock*	A reference to a mutex lock.

Returns

A reference to the copy of *mutex_lock*.

5.29.4 Member Data Documentation

5.29.4.1 m_mutex

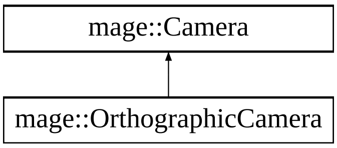
```
Mutex& mage::MutexLock::m_mutex  [private]
```

The mutex of this mutex lock.

5.30 mage::OrthographicCamera Class Reference

```
#include <orthographic_camera.hpp>
```

Inheritance diagram for mage::OrthographicCamera:



Private Member Functions

- OrthographicCamera (float width, float height, float near_z=MAGE_DEFAULT_CAMERA_NEAR_Z, float far_z=MAGE_DEFAULT_CAMERA_FAR_Z)
- OrthographicCamera (const OrthographicCamera &camera)
- virtual ∼OrthographicCamera ()
- virtual XMMATRIX GetViewToProjectionMatrix () const override
- void SetViewToProjectionMatrix (float width, float height, float near_z=MAGE_DEFAULT_CAMERA_NEA↩R_Z, float far_z=MAGE_DEFAULT_CAMERA_FAR_Z)

Additional Inherited Members

5.30.1 Detailed Description

A class of orthographic cameras.

5.30.2 Constructor & Destructor Documentation

5.30.2.1 OrthographicCamera() [1/2]

```
mage::OrthographicCamera::OrthographicCamera (
            float width,
            float height,
            float near_z = MAGE_DEFAULT_CAMERA_NEAR_Z,
            float far_z = MAGE_DEFAULT_CAMERA_FAR_Z )  [private]
```

Constructs an orthographic camera.

Parameters

in	*width*	The width.
in	*height*	The height.
in	*near↩_z*	The position of the near z-plane.
in	*far_z*	The position of the far z-plane.

5.30.2.2 OrthographicCamera() [2/2]

```
mage::OrthographicCamera::OrthographicCamera (
             const OrthographicCamera & camera )  [private]
```

Constructs an orthographic camera from the given orthographic camera.

Parameters

in	*camera*	A reference to the orthographic camera.

5.30.2.3 ∼OrthographicCamera()

```
virtual mage::OrthographicCamera::∼OrthographicCamera ( )  [private], [virtual]
```

Destructs this orthographic camera.

5.30.3 Member Function Documentation

5.30.3.1 GetViewToProjectionMatrix()

```
virtual XMMATRIX mage::OrthographicCamera::GetViewToProjectionMatrix ( ) const  [override],
[private], [virtual]
```

Returns the view-to-projection matrix of this orthographic camera.

Returns

The view-to-projection matrix of this orthographic camera.

Implements mage::Camera.

5.30.3.2 SetViewToProjectionMatrix()

```
void mage::OrthographicCamera::SetViewToProjectionMatrix (
             float width,
             float height,
             float near_z = MAGE_DEFAULT_CAMERA_NEAR_Z,
             float far_z = MAGE_DEFAULT_CAMERA_FAR_Z )  [private]
```

Sets the view-to-projection matrix of this orthographic camera.

Parameters

in	*width*	The width.
in	*height*	The height.
in	*near← _z*	The position of the near z-plane.
in	*far_z*	The position of the far z-plane.

5.31 mage::ParallelForLoop Class Reference

Public Member Functions

- ParallelForLoop (function< void(size_t) > func, size_t max_index, size_t chunk_size)
- bool IsFinished () const

Public Attributes

- function< void(size_t) > func
- size_t m_next_index
- const size_t m_max_index
- const size_t m_chunk_size
- ParallelForLoop ∗ m_next
- size_t m_active_workers

5.31.1 Constructor & Destructor Documentation

5.31.1.1 ParallelForLoop()

```
mage::ParallelForLoop::ParallelForLoop (
             function< void(size_t) > func,
             size_t max_index,
             size_t chunk_size )
```

5.31.2 Member Function Documentation

5.31.2.1 IsFinished()

```
bool mage::ParallelForLoop::IsFinished ( ) const
```

5.31.3 Member Data Documentation

5.31.3.1 func

```
function< void(size_t) > mage::ParallelForLoop::func
```

5.31.3.2 m_active_workers

```
size_t mage::ParallelForLoop::m_active_workers
```

5.31.3.3 m_chunk_size

```
const size_t mage::ParallelForLoop::m_chunk_size
```

5.31.3.4 m_max_index

```
const size_t mage::ParallelForLoop::m_max_index
```

5.31.3.5 m_next

```
ParallelForLoop* mage::ParallelForLoop::m_next
```

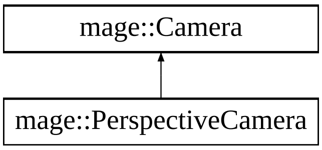
5.31.3.6 m_next_index

```
size_t mage::ParallelForLoop::m_next_index
```

5.32 mage::PerspectiveCamera Class Reference

```
#include <perspective_camera.hpp>
```

Inheritance diagram for mage::PerspectiveCamera:



Public Member Functions

- PerspectiveCamera (float width, float height, float fov_y=MAGE_DEFAULT_CAMERA_FOV_Y, float near_↩ z=MAGE_DEFAULT_CAMERA_NEAR_Z, float far_z=MAGE_DEFAULT_CAMERA_FAR_Z)
- PerspectiveCamera (const PerspectiveCamera &camera)
- virtual ~PerspectiveCamera ()
- float GetFOVY () const
- Camera & SetFOVY (float fov_y)
- virtual XMMATRIX GetViewToProjectionMatrix () const override
- void SetViewToProjectionMatrix (float width, float height, float fov_y=MAGE_DEFAULT_CAMERA_FOV_Y, float near_z=MAGE_DEFAULT_CAMERA_NEAR_Z, float far_z=MAGE_DEFAULT_CAMERA_FAR_Z)

Protected Attributes

- float m_fov_y

Additional Inherited Members

5.32.1 Detailed Description

A class of perspective camera.

5.32.2 Constructor & Destructor Documentation

5.32.2.1 PerspectiveCamera() [1/2]

```
mage::PerspectiveCamera::PerspectiveCamera (
            float width,
            float height,
            float fov_y = MAGE_DEFAULT_CAMERA_FOV_Y,
            float near_z = MAGE_DEFAULT_CAMERA_NEAR_Z,
            float far_z = MAGE_DEFAULT_CAMERA_FAR_Z )
```

Constructs a perspective camera.

Parameters

in	*width*	The width.
in	*height*	The height.
in	*fov_y*	The vertical field-of-view.
in	*near_z*	The position of the near z-plane.
in	*far_z*	The position of the far z-plane.

5.32.2.2 PerspectiveCamera() [2/2]

```
mage::PerspectiveCamera::PerspectiveCamera (
            const PerspectiveCamera & camera )
```

Constructs a perspective camera from the given perpsective camera.

Parameters

in	*camera*	A reference to the perspective camera.

5.32.2.3 ∼PerspectiveCamera()

```
virtual mage::PerspectiveCamera::∼PerspectiveCamera ( )  [virtual]
```

Destructs this perspective camera.

5.32.3 Member Function Documentation

5.32.3.1 GetFOVY()

```
float mage::PerspectiveCamera::GetFOVY ( ) const
```

Returns the vertical field-of-view of this perspective camera.

Returns

The vertical field-of-view of this perspective camera.

5.32.3.2 GetViewToProjectionMatrix()

```
virtual XMMATRIX mage::PerspectiveCamera::GetViewToProjectionMatrix ( ) const  [override],
[virtual]
```

Returns the view-to-projection matrix of this perspective camera.

Returns

The view-to-projection matrix of this perspective camera.

Implements mage::Camera.

5.32.3.3 SetFOVY()

```
Camera& mage::PerspectiveCamera::SetFOVY (
             float fov_y )
```

Sets the vertical field-of-view of this perspective camera to the given value.

Parameters

in	*fov_y*	The vertical field-of-view.

Returns

A reference to this perspective camera.

5.32.3.4 SetViewToProjectionMatrix()

```
void mage::PerspectiveCamera::SetViewToProjectionMatrix (
             float width,
             float height,
             float fov_y = MAGE_DEFAULT_CAMERA_FOV_Y,
             float near_z = MAGE_DEFAULT_CAMERA_NEAR_Z,
             float far_z = MAGE_DEFAULT_CAMERA_FAR_Z )
```

Sets the view-to-projection matrix of this perspective camera.

Parameters

in	*width*	The width.
in	*height*	The height.
in	*fov_y*	The vertical field-of-view.
in	*near_z*	The position of the near z-plane.
in	*far_z*	The position of the far z-plane.

5.32.4 Member Data Documentation

5.32.4.1 m_fov_y

```
float mage::PerspectiveCamera::m_fov_y  [protected]
```

The vertical field-of-view of this perspective camera.

5.33 mage::ProgressReporter Class Reference

```
#include <progressreporter.hpp>
```

Public Member Functions

- ProgressReporter (uint32_t nb_work, const string &title, uint32_t bar_length=0)
- virtual ∼ProgressReporter ()
- void Update (uint32_t nb_work=1)
- void Done ()

Protected Attributes

- const uint32_t m_nb_work_total
- uint32_t m_nb_work_done
- uint32_t m_nb_plusses_total
- uint32_t m_nb_plusses_printed
- Timer ∗ m_timer
- FILE ∗ m_fout
- char ∗ m_buffer
- char ∗ m_current_pos
- Mutex ∗ m_mutex

5.33.1 Detailed Description

A class of progress reporters.

5.33.2 Constructor & Destructor Documentation

5.33.2.1 ProgressReporter()

```
mage::ProgressReporter::ProgressReporter (
            uint32_t nb_work,
            const string & title,
            uint32_t bar_length = 0 )
```

Constructs a progress reporter.

Parameters

in	*nb_work*	The number of parts of the total work.
in	*title*	A reference to the title.
in	*bar_length*	The length of the progress bar. If 0 the default length will be chosen.

5.33.2.2 ∼ProgressReporter()

```
mage::ProgressReporter::∼ProgressReporter ( ) [virtual]
```

Destructs this progress reporter.

5.33.3 Member Function Documentation

5.33.3.1 Done()

```
void mage::ProgressReporter::Done ( )
```

Finishes this progress reporter.

5.33.3.2 Update()

```
void mage::ProgressReporter::Update (
             uint32_t nb_work = 1 )
```

Updates this progress reporter.

Parameters

in	*nb_work*	The number of parts of the total work that are done.

5.33.4 Member Data Documentation

5.33.4.1 m_buffer

```
char* mage::ProgressReporter::m_buffer [protected]
```

The output buffer of this progress reporter.

5.33.4.2 m_current_pos

```
char* mage::ProgressReporter::m_current_pos [protected]
```

The current (output) position of this progress reporter.

5.33.4.3 m_fout

```
FILE* mage::ProgressReporter::m_fout  [protected]
```

The output file stream of this progress reporter.

5.33.4.4 m_mutex

```
Mutex* mage::ProgressReporter::m_mutex  [protected]
```

The mutex needed for updating this progress reporter.

5.33.4.5 m_nb_plusses_printed

```
uint32_t mage::ProgressReporter::m_nb_plusses_printed  [protected]
```

The total number of plusses that are already outputted.

5.33.4.6 m_nb_plusses_total

```
uint32_t mage::ProgressReporter::m_nb_plusses_total  [protected]
```

The total number of plusses to output.

5.33.4.7 m_nb_work_done

```
uint32_t mage::ProgressReporter::m_nb_work_done  [protected]
```

The number of parts of the total work that are already done.

5.33.4.8 m_nb_work_total

```
const uint32_t mage::ProgressReporter::m_nb_work_total  [protected]
```

The number of parts of the total work.

5.33.4.9 m_timer

```
Timer* mage::ProgressReporter::m_timer  [protected]
```

The timer of this progress reporter.

5.34 mage::ReadWriteMutex Class Reference

```
#include <lock.hpp>
```

Static Public Member Functions

- static ReadWriteMutex ∗ Create ()
- static void Destroy (ReadWriteMutex ∗mutex)

Private Member Functions

- ReadWriteMutex ()
- ReadWriteMutex (ReadWriteMutex &mutex)
- ∼ReadWriteMutex ()
- ReadWriteMutex & operator= (const ReadWriteMutex &mutex)
- void AcquireRead ()
- void ReleaseRead ()
- void AcquireWrite ()
- void ReleaseWrite ()

Private Attributes

- LONG m_nb_writers_waiting
- LONG m_nb_readers_waiting
- DWORD m_active_writer_readers
- HANDLE m_ready_to_read_handle
- HANDLE m_ready_to_write_handle
- CRITICAL_SECTION m_critical_section

Friends

- struct ReadWriteMutexLock

5.34.1 Detailed Description

A class of read write mutexes.

5.34.2 Constructor & Destructor Documentation

5.34.2.1 ReadWriteMutex() [1/2]

```
mage::ReadWriteMutex::ReadWriteMutex ( )  [private]
```

Constructs a read write mutex.

5.34.2.2 ReadWriteMutex() [2/2]

```
mage::ReadWriteMutex::ReadWriteMutex (
             ReadWriteMutex & mutex )  [private]
```

Constructs a read write mutex from the given read write mutex.

Parameters

in	*mutex*	The read write mutex.

5.34.2.3 ∼ReadWriteMutex()

```
mage::ReadWriteMutex::∼ReadWriteMutex ( )  [private]
```

Destructs this read write mutex.

5.34.3 Member Function Documentation

5.34.3.1 AcquireRead()

```
void mage::ReadWriteMutex::AcquireRead ( )  [private]
```

Acquires a read.

5.34.3.2 AcquireWrite()

```
void mage::ReadWriteMutex::AcquireWrite ( )  [private]
```

Acquires a write.

5.34.3.3 Create()

```
static ReadWriteMutex* mage::ReadWriteMutex::Create ( )  [static]
```

Creates a mutex.

5.34.3.4 Destroy()

```
static void mage::ReadWriteMutex::Destroy (
             ReadWriteMutex * mutex )  [static]
```

Destroys a given read write mutex.

Parameters

in	*mutex*	The read write mutex to destroy.

5.34.3.5 operator=()

```
ReadWriteMutex& mage::ReadWriteMutex::operator= (
             const ReadWriteMutex & mutex )  [private]
```

Copies the given read write mutex to this read write mutex.

Parameters

in	*mutex*	A reference to a read write mutex.

Returns

A reference to the copy of *mutex*.

5.34.3.6 ReleaseRead()

```
void mage::ReadWriteMutex::ReleaseRead ( )  [private]
```

Release a read.

5.34.3.7 ReleaseWrite()

```
void mage::ReadWriteMutex::ReleaseWrite ( )  [private]
```

Release a write.

5.34.4 Friends And Related Function Documentation

5.34.4.1 ReadWriteMutexLock

```
friend struct ReadWriteMutexLock  [friend]
```

5.34.5 Member Data Documentation

5.34.5.1 m_active_writer_readers

```
DWORD mage::ReadWriteMutex::m_active_writer_readers  [private]
```

The active group of this read write mutex lock.

HIWORD is the flag indicating a writer is active. LOWORD is the number of active readers.

5.34.5.2 m_critical_section

```
CRITICAL_SECTION mage::ReadWriteMutex::m_critical_section  [private]
```

The critical section object of this read write mutex.

5.34.5.3 m_nb_readers_waiting

```
LONG mage::ReadWriteMutex::m_nb_readers_waiting  [private]
```

The number of readers waiting for this read write mutex lock.

5.34.5.4 m_nb_writers_waiting

```
LONG mage::ReadWriteMutex::m_nb_writers_waiting  [private]
```

The number of writers waiting for this read write mutex lock.

5.34.5.5 m_ready_to_read_handle

```
HANDLE mage::ReadWriteMutex::m_ready_to_read_handle  [private]
```

The handle of this read write mutex lock if ready for reading.

5.34.5.6 m_ready_to_write_handle

```
HANDLE mage::ReadWriteMutex::m_ready_to_write_handle  [private]
```

The handle of this read write mutex lock if ready for writing.

5.35 mage::ReadWriteMutexLock Struct Reference

```
#include <lock.hpp>
```

Public Member Functions

- ReadWriteMutexLock (ReadWriteMutex &mutex, ReadWriteMutexLockType lock_type)
- ~ReadWriteMutexLock ()
- void UpgradeToWrite ()
- void DowngradeToRead ()

Private Member Functions

- ReadWriteMutexLock (const ReadWriteMutexLock &mutex_lock)
- ReadWriteMutexLock & operator= (const ReadWriteMutexLock &mutex_lock)

Private Attributes

- ReadWriteMutexLockType m_type
- ReadWriteMutex & m_mutex

5.35.1 Detailed Description

A struct of read write mutex locks.

5.35.2 Constructor & Destructor Documentation

5.35.2.1 ReadWriteMutexLock() [1/2]

```
mage::ReadWriteMutexLock::ReadWriteMutexLock (
            ReadWriteMutex & mutex,
            ReadWriteMutexLockType lock_type )
```

Constructs a read write mutex lock for the given read write mutex and lock type.

Parameters

in	*mutex*	A reference to a read write mutex.
in	*lock_type*	The lock type.

5.35.2.2 ∼ReadWriteMutexLock()

```
mage::ReadWriteMutexLock::∼ReadWriteMutexLock ( )
```

Destructs this read write mutex lock.

5.35.2.3 ReadWriteMutexLock() [2/2]

```
mage::ReadWriteMutexLock::ReadWriteMutexLock (
            const ReadWriteMutexLock & mutex_lock )   [private]
```

Constructs a read write mutex lock from the given read write mutex lock.

Parameters

in	*mutex_lock*	A reference to a read write mutex lock.

5.35.3 Member Function Documentation

5.35.3.1 DowngradeToRead()

```
void mage::ReadWriteMutexLock::DowngradeToRead ( )
```

Downgrades this read write lock to read.

5.35.3.2 operator=()

```
ReadWriteMutexLock& mage::ReadWriteMutexLock::operator= (
             const ReadWriteMutexLock & mutex_lock )  [private]
```

Copies the given read write mutex lock to this read write mutex lock.

Parameters

in	*mutex_lock*	A reference to a read write mutex lock.

Returns

A reference to the copy of *mutex_lock*.

5.35.3.3 UpgradeToWrite()

```
void mage::ReadWriteMutexLock::UpgradeToWrite ( )
```

Upgrades this read write lock to write.

5.35.4 Member Data Documentation

5.35.4.1 m_mutex

```
ReadWriteMutex& mage::ReadWriteMutexLock::m_mutex  [private]
```

The read write mutex of this read write mutex lock.

5.35.4.2 m_type

```
ReadWriteMutexLockType mage::ReadWriteMutexLock::m_type  [private]
```

The lock type of this read write mutex lock.

5.36 mage::Reference< T > Class Template Reference

```
#include <reference.hpp>
```

Public Member Functions

- Reference (T ∗ptr=NULL)
- Reference (const Reference< T > &reference)
- virtual ∼Reference ()
- Reference & operator= (T ∗ptr)
- Reference & operator= (const Reference< T > &reference)
- T ∗ operator-> ()
- const T ∗ operator-> () const
- const T ∗ GetPtr () const
- operator bool () const

Private Attributes

- T ∗ m_ptr

5.36.1 Detailed Description

template<typename T>
class mage::Reference< T >

A class of references.

Template Parameters

T	The type of reference.

5.36.2 Constructor & Destructor Documentation

5.36.2.1 Reference() [1/2]

```
template<typename T>
mage::Reference< T >::Reference (
            T ∗ ptr = NULL )
```

Constructs a reference for the given pointer.

Parameters

in	*ptr*	The pointer.

5.36.2.2 Reference() [2/2]

```
template<typename T>
mage::Reference< T >::Reference (
            const Reference< T > & reference )
```

Constructs a reference from the given reference.

Parameters

in	*reference*	The reference.

5.36.2.3 ∼Reference()

```
template<typename T>
virtual mage::Reference< T >::∼Reference ( )  [virtual]
```

Destructs this reference.

5.36.3 Member Function Documentation

5.36.3.1 GetPtr()

```
template<typename T>
const T* mage::Reference< T >::GetPtr ( ) const
```

Returns the pointer of this reference.

Returns

The pointer of this reference.

5.36.3.2 operator bool()

```
template<typename T>
mage::Reference< T >::operator bool ( ) const
```

Checks whether the pointer of this reference does not point to `NULL`.

Returns

`true` if the pointer of this reference does not point to `NULL`. `false` otherwise.

5.36.3.3 operator->() [1/2]

```
template<typename T>
T* mage::Reference< T >::operator-> ( )
```

Dereferences this reference.

Returns

The pointer of this reference.

5.36.3.4 operator->() [2/2]

```
template<typename T>
const T* mage::Reference< T >::operator-> ( ) const
```

Dereferences this reference.

Returns

The pointer of this reference.

5.36.3.5 operator=() [1/2]

```
template<typename T>
Reference& mage::Reference< T >::operator= (
              T * ptr )
```

Copies the given pointer into a reference.

Parameters

in	*ptr*	The pointer.

Returns

A reference for *ptr*.

5.36.3.6 operator=() [2/2]

```
template<typename T>
Reference& mage::Reference< T >::operator= (
                const Reference< T > & reference )
```

Copies the given reference into a reference.

Parameters

in	*reference*	The reference.

Returns

A reference for *reference*.

5.36.4 Member Data Documentation

5.36.4.1 m_ptr

```
template<typename T>
T* mage::Reference< T >::m_ptr  [private]
```

The pointer of this reference.

5.37 mage::ReferenceCounted Class Reference

```
#include <reference.hpp>
```

Public Member Functions

- uint32_t IncrementReferenceCount ()
- uint32_t DecrementReferenceCount ()

Protected Member Functions

- ReferenceCounted ()

Private Attributes

- AtomicInt32 m_reference_count

5.37.1 Detailed Description

A class of reference counted objects.

5.37.2 Constructor & Destructor Documentation

5.37.2.1 ReferenceCounted()

```
mage::ReferenceCounted::ReferenceCounted ( )  [protected]
```

Constructs a reference counted object.

5.37.3 Member Function Documentation

5.37.3.1 DecrementReferenceCount()

```
uint32_t mage::ReferenceCounted::DecrementReferenceCount ( )
```

Decrements the reference count of this reference counted object.

Returns

The final reference count of this reference counted object.

5.37.3.2 IncrementReferenceCount()

```
uint32_t mage::ReferenceCounted::IncrementReferenceCount ( )
```

Increments the reference count of this reference counted object.

Returns

The final reference count of this reference counted object.

5.37.4 Member Data Documentation

5.37.4.1 m_reference_count

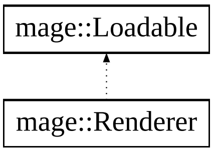
```
AtomicInt32 mage::ReferenceCounted::m_reference_count  [private]
```

The reference count of this reference counted object.

5.38 mage::Renderer Class Reference

```
#include <renderer.hpp>
```

Inheritance diagram for mage::Renderer:



Public Member Functions

- bool IsWindowed () const
- bool IsFullScreen () const
- bool LostMode () const
- void SwitchMode (bool toggle)

Protected Member Functions

- Renderer (HWND hwindow)
- virtual ~Renderer ()
- HRESULT InitializeRenderer ()
- HRESULT UnitializeRenderer ()
- HRESULT SetupDevice ()
- HRESULT SetupSwapChain ()
- HRESULT SetupRenderTargetView ()
- HRESULT SetupDepthStencilView ()
- HRESULT SetupViewPort ()
- void Render (double elapsed_time)

Protected Attributes

- HWND m_hwindow
- bool m_fullscreen
- D3D_FEATURE_LEVEL m_feature_level
- ID3D11Device2 ∗ m_device2
- ID3D11DeviceContext2 ∗ m_device_context2
- IDXGISwapChain2 ∗ m_swap_chain2
- ID3D11RenderTargetView ∗ m_render_target_view
- ID3D11Texture2D ∗ m_depth_stencil
- ID3D11DepthStencilView ∗ m_depth_stencil_view

Friends

- class Engine

Additional Inherited Members

5.38.1 Detailed Description

A class of renderers.

5.38.2 Constructor & Destructor Documentation

5.38.2.1 Renderer()

```
mage::Renderer::Renderer (
            HWND hwindow )  [protected]
```

Constructs a renderer.

Parameters

in	*hwindow*	The main window handle.

5.38.2.2 ∼Renderer()

```
mage::Renderer::∼Renderer ( )  [protected], [virtual]
```

Destructs this renderer.

5.38.3 Member Function Documentation

5.38.3.1 InitializeRenderer()

```
HRESULT mage::Renderer::InitializeRenderer ( )  [protected]
```

Initializes this renderer.

Returns

A success/error value.

5.38.3.2 IsFullScreen()

```
bool mage::Renderer::IsFullScreen ( ) const
```

Checks whether this renderer renders in full screen mode.

Returns

`true` if this renderer renders in full screen mode. `false` otherwise.

5.38.3.3 IsWindowed()

```
bool mage::Renderer::IsWindowed ( ) const
```

Checks whether this renderer renders in windowed mode.

Returns

`true` if this renderer renders in windowed mode. `false` otherwise.

5.38.3.4 LostMode()

```
bool mage::Renderer::LostMode ( ) const
```

Checks whether this renderer lost its mode, i.e. the current mode of this renderer differs from the cyrrent mode of its swap chain (due to for example ALT + TAB).

5.38.3.5 Render()

```
void mage::Renderer::Render (
            double elapsed_time )  [protected]
```

Renders the current frame.

Parameters

in	*elapsed_time*	The elapsed time since the previous frame.

5.38.3.6 SetupDepthStencilView()

```
HRESULT mage::Renderer::SetupDepthStencilView ( )  [protected]
```

Sets up the depth stencil view of this renderer.

Returns

A success/error value.

5.38.3.7 SetupDevice()

```
HRESULT mage::Renderer::SetupDevice ( )  [protected]
```

Setup the D3D11 device and context of this renderer.

Returns

A success/error value.

5.38.3.8 SetupRenderTargetView()

```
HRESULT mage::Renderer::SetupRenderTargetView ( ) [protected]
```

Sets up the render target view of this renderer.

Returns

A success/error value.

5.38.3.9 SetupSwapChain()

```
HRESULT mage::Renderer::SetupSwapChain ( ) [protected]
```

Sets up the swap chain of this renderer.

Returns

A success/error value.

5.38.3.10 SetupViewPort()

```
HRESULT mage::Renderer::SetupViewPort ( ) [protected]
```

Sets up and binds the viewport of this renderer to the graphics pipeline.

Returns

A success/error value.

5.38.3.11 SwitchMode()

```
void mage::Renderer::SwitchMode (
            bool toggle )
```

Recreates the swap chain buffers and switches the mode of this renderer. Windowed mode is switched to full screen mode and vice versa.

Returns

toggle If `true` only the swap chain buffers will be recreated to match the current mode of the swap chain and no mode switch will occurs. If `false` both the swap chain buffers will be replaced and a mode switch will occur.

5.38.3.12 UnitializeRenderer()

```
HRESULT mage::Renderer::UnitializeRenderer ( )  [protected]
```

Uninitializes this renderer.

Returns

A success/error value.

5.38.4 Friends And Related Function Documentation

5.38.4.1 Engine

```
friend class Engine  [friend]
```

5.38.5 Member Data Documentation

5.38.5.1 m_depth_stencil

```
ID3D11Texture2D* mage::Renderer::m_depth_stencil  [protected]
```

5.38.5.2 m_depth_stencil_view

```
ID3D11DepthStencilView* mage::Renderer::m_depth_stencil_view  [protected]
```

5.38.5.3 m_device2

```
ID3D11Device2* mage::Renderer::m_device2  [protected]
```

5.38.5.4 m_device_context2

```
ID3D11DeviceContext2* mage::Renderer::m_device_context2  [protected]
```

5.38.5.5 m_feature_level

```
D3D_FEATURE_LEVEL mage::Renderer::m_feature_level  [protected]
```

5.38.5.6 m_fullscreen

```
bool mage::Renderer::m_fullscreen  [protected]
```

A flag indicating whether this renderer uses a full screen mode (if `true`) or a windowed mode (if `false`).

5.38.5.7 m_hwindow

```
HWND mage::Renderer::m_hwindow [protected]
```

Main window handle of this renderer.

5.38.5.8 m_render_target_view

```
ID3D11RenderTargetView* mage::Renderer::m_render_target_view [protected]
```

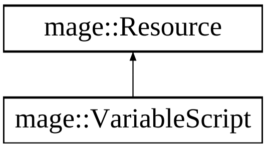
5.38.5.9 m_swap_chain2

```
IDXGISwapChain2* mage::Renderer::m_swap_chain2 [protected]
```

5.39 mage::Resource Class Reference

```
#include <resource.hpp>
```

Inheritance diagram for mage::Resource:



Public Member Functions

- Resource (const string &name, const string &path="./")
- virtual ~Resource ()
- const string & GetName () const
- const string & GetPath () const
- const string GetFilename () const

Private Member Functions

- uint32_t IncrementResourceReferenceCount ()
- uint32_t DecrementResourceReferenceCount ()

Private Attributes

- AtomicInt32 m_resource_reference_count
- const string m_name
- const string m_path

Friends

- template<typename T >
  class ResourceManager

5.39.1 Detailed Description

A class of resources.

5.39.2 Constructor & Destructor Documentation

5.39.2.1 Resource()

```
mage::Resource::Resource (
             const string & name,
             const string & path = "./" )
```

Constructs a resource with a given name and path.

Parameters

in	*name*	A reference to the name.
in	*path*	A reference to the path.

5.39.2.2 ∼Resource()

```
virtual mage::Resource::∼Resource ( )  [virtual]
```

Destructs this resource.

5.39.3 Member Function Documentation

5.39.3.1 DecrementResourceReferenceCount()

```
uint32_t mage::Resource::DecrementResourceReferenceCount ( )  [private]
```

Decrements the resource reference count of this reference counted object.

Returns

The final resource reference count of this reference counted object.

5.39.3.2 GetFilename()

```
const string mage::Resource::GetFilename ( ) const
```

Returns the filename of this resource.

Returns

The filename of this resource.

5.39.3.3 GetName()

```
const string& mage::Resource::GetName ( ) const
```

Returns the name of this resource.

Returns

A reference to the name of this resource.

5.39.3.4 GetPath()

```
const string& mage::Resource::GetPath ( ) const
```

Returns the path of this resource.

Returns

A reference to the path of this resource.

5.39.3.5 IncrementResourceReferenceCount()

```
uint32_t mage::Resource::IncrementResourceReferenceCount ( ) [private]
```

Increments the resource reference count of this reference counted object.

Returns

The final resource reference count of this reference counted object.

5.39.4 Friends And Related Function Documentation

5.39.4.1 ResourceManager

```
template<typename T >
friend class ResourceManager [friend]
```

5.39.5 Member Data Documentation

5.39.5.1 m_name

```
const string mage::Resource::m_name  [private]
```

The name of this resource.

5.39.5.2 m_path

```
const string mage::Resource::m_path  [private]
```

The path of this resource.

5.39.5.3 m_resource_reference_count

```
AtomicInt32 mage::Resource::m_resource_reference_count  [private]
```

The resource reference count of this resource.

5.40 mage::ResourceManager< T > Class Template Reference

```
#include <resource_manager.hpp>
```

Public Member Functions

- ResourceManager (void(∗CreateResourceFunction)(T ∗∗resource, const string &name, const string &path)=nullptr)
- virtual ∼ResourceManager ()
- T ∗ AddResource (const string &name, const string &path="./")
- void RemoveResource (T ∗resource)
- void ClearResources ()
- T ∗ GetResource (const string &name, const string &path="./") const

Protected Attributes

- list< T ∗> m_resources
- void(∗ CreateResource)(T ∗∗resource, const string &name, const string &path)

5.40.1 Detailed Description

template<typename T>
class mage::ResourceManager< T >

A class of resource managers.

Template Parameters

T	The type of resources.

5.40.2 Constructor & Destructor Documentation

5.40.2.1 ResourceManager()

```
template<typename T>
mage::ResourceManager< T >::ResourceManager (
              void(*)(T **resource, const string &name, const string &path) CreateResource↩
Function = nullptr )
```

Constructs a resource manager.

Parameters

in	*CreateResourceFunction*	The application specific resource creation function.

5.40.2.2 ∼ResourceManager()

```
template<typename T>
virtual mage::ResourceManager< T >::∼ResourceManager ( )  [virtual]
```

Destructs this resource manager.

5.40.3 Member Function Documentation

5.40.3.1 AddResource()

```
template<typename T>
T* mage::ResourceManager< T >::AddResource (
              const string & name,
              const string & path = "./" )
```

Adds a new resource to this resource manager.

Parameters

in	*name*	A reference to the name of the new resource.
in	*path*	A reference to the path of the new resource.

Returns

A pointer to the resource.

5.40.3.2 ClearResources()

```
template<typename T>
void mage::ResourceManager< T >::ClearResources ( )
```

Removes and destructs all the resources from this resource manager, and leaving the resource manager with no resources.

5.40.3.3 GetResource()

```
template<typename T>
T* mage::ResourceManager< T >::GetResource (
            const string & name,
            const string & path = "./" ) const
```

Returns a resource of this resource manager by its filename (given name and path).

Parameters

in	*name*	A reference to the name of the new resource.
in	*path*	A reference to the path of the new resource.

Returns

`nullptr` if the resource is not present.
A pointer to the resource.

5.40.3.4 RemoveResource()

```
template<typename T>
void mage::ResourceManager< T >::RemoveResource (
            T * resource )
```

Removes the given resource from this resource manager.

Parameters

in,out	*resource*	A pointer to the resource.

5.40.4 Member Data Documentation

5.40.4.1 CreateResource

```
template<typename T>
void(* mage::ResourceManager< T >::CreateResource) (T **resource, const string &name, const
string &path)  [protected]
```

The application specific resource creation function for the resources of this resource manager.

5.40.4.2 m_resources

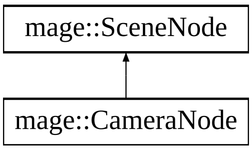
```
template<typename T>
list< T * > mage::ResourceManager< T >::m_resources  [protected]
```

The linked list containing the resources of this resource manager.

5.41 mage::SceneNode Class Reference

```
#include <scene_node.hpp>
```

Inheritance diagram for mage::SceneNode:



Public Member Functions

- virtual ∼SceneNode ()
- SceneNode ∗ GetParent () const
- bool ContainsChild (const SceneNode ∗child) const
- void AddChild (SceneNode ∗child)
- void RemoveChild (SceneNode ∗child)
- size_t GetNbOfChilds () const
- Transform & GetTransform ()
- const Transform & GetTransform () const
- XMMATRIX GetParentToObjectMatrix () const
- XMMATRIX GetParentToWorldMatrix () const
- XMMATRIX GetWorldToObjectMatrix () const
- XMMATRIX GetObjectToWorldMatrix () const
- virtual void Accept (SceneNodeVisitor &visitor)=0
- virtual void Accept (SceneNodeVisitor &visitor) const =0

Protected Member Functions

- SceneNode (const Transform &transform=Transform())
- void PassToChilds (SceneNodeVisitor &visitor)
- void PassToChilds (SceneNodeVisitor &visitor) const

Private Member Functions

- void SetParent (SceneNode ∗parent)

Private Attributes

- Transform m_transform
- SceneNode ∗ m_parent
- set< SceneNode ∗, std::less<> > m_childs

5.41.1 Detailed Description

A class of scene nodes.

5.41.2 Constructor & Destructor Documentation

5.41.2.1 ∼SceneNode()

```
virtual mage::SceneNode::∼SceneNode ( )  [virtual]
```

Destructs this scene node.

5.41.2.2 SceneNode()

```
mage::SceneNode::SceneNode (
             const Transform & transform = Transform() )  [protected]
```

Constructs a scene node with the given transform.

Parameters

in	*transform*	A reference to the transform.

5.41.3 Member Function Documentation

5.41.3.1 Accept() [1/2]

```
virtual void mage::SceneNode::Accept (
             SceneNodeVisitor & visitor )  [pure virtual]
```

Accepts the given visitor.

Parameters

in	*visitor*	A reference to the visitor.

Implemented in mage::CameraNode.

5.41.3.2 Accept() [2/2]

```
virtual void mage::SceneNode::Accept (
             SceneNodeVisitor & visitor ) const  [pure virtual]
```

Accepts the given visitor.

Parameters

in	*visitor*	A reference to the visitor.

Implemented in mage::CameraNode.

5.41.3.3 AddChild()

```
void mage::SceneNode::AddChild (
             SceneNode * child )
```

Adds the given child scene node to the child scene nodes of this scene node. If the given child scene node has already a parent scene node, it is removed from that node since scene nodes may only have at most one parent scene node.

Parameters

in	*child*	A pointer to the child scene node.

5.41.3.4 ContainsChild()

```
bool mage::SceneNode::ContainsChild (
             const SceneNode * child ) const
```

Checks whether this scene node contains the given scene node as a child scene node.

Returns

`true` if this scene node contains the given scene node as a child scene node. `false` otherwise.

5.41.3.5 GetNbOfChilds()

```
size_t mage::SceneNode::GetNbOfChilds ( ) const
```

Returns the total number of child scene nodes of this scene node.

Returns

The total number of child scene nodes of this scene node.

5.41.3.6 GetObjectToWorldMatrix()

```
XMMATRIX mage::SceneNode::GetObjectToWorldMatrix ( ) const
```

Returns the object-to-world matrix of this scene node.

Returns

The object-to-world matrix of this scene node.

5.41.3.7 GetParent()

```
SceneNode* mage::SceneNode::GetParent ( ) const
```

Returns the parent scene node of this scene node.

Returns

`nullptr` if this scene node has no parent scene node (i.e. this scene node is a root node).
A pointer to the parent scene node of this scene node.

5.41.3.8 GetParentToObjectMatrix()

```
XMMATRIX mage::SceneNode::GetParentToObjectMatrix ( ) const
```

Returns the parent-to-object matrix of this scene node.

Returns

The parent-to-object matrix of this scene node.

5.41.3.9 GetParentToWorldMatrix()

```
XMMATRIX mage::SceneNode::GetParentToWorldMatrix ( ) const
```

Returns the object-to-parent matrix of this scene node.

Returns

The object-to-parent matrix of this scene node.

5.41.3.10 GetTransform() [1/2]

```
Transform& mage::SceneNode::GetTransform ( )
```

Returns the transform of this scene node.

Returns

The transform of this scene node.

5.41.3.11 GetTransform() [2/2]

```
const Transform& mage::SceneNode::GetTransform ( ) const
```

Returns the transform of this scene node.

Returns

The transform of this scene node.

5.41.3.12 GetWorldToObjectMatrix()

```
XMMATRIX mage::SceneNode::GetWorldToObjectMatrix ( ) const
```

Returns the world-to-object matrix of this scene node.

Returns

The world-to-object matrix of this scene node.

5.41.3.13 PassToChilds() [1/2]

```
void mage::SceneNode::PassToChilds (
             SceneNodeVisitor & visitor )   [protected]
```

Pass the given visitor to the childs of this scene node.

Parameters

in	*visitor*	A reference to the visitor.

5.41.3.14 PassToChilds() [2/2]

```
void mage::SceneNode::PassToChilds (
             SceneNodeVisitor & visitor ) const   [protected]
```

Pass the given visitor to the childs of this scene node.

Parameters

in	*visitor*	A reference to the visitor.

5.41.3.15 RemoveChild()

```
void mage::SceneNode::RemoveChild (
             SceneNode * child )
```

Removes the given child scene node from the child scene nodes of this scene node.

Parameters

in	*child*	A pointer to the child scene node.

5.41.3.16 SetParent()

```
void mage::SceneNode::SetParent (
            SceneNode * parent )  [private]
```

Sets the parent scene node of this scene node to the given scene node.

Parameters

in	*parent*	A pointer to the parent scene node.

5.41.4 Member Data Documentation

5.41.4.1 m_childs

```
set< SceneNode *, std::less<> > mage::SceneNode::m_childs  [private]
```

A set containing the child scene nodes of this scene node.

5.41.4.2 m_parent

```
SceneNode* mage::SceneNode::m_parent  [private]
```

A pointer to the parent scene node of this scene node.

5.41.4.3 m_transform

```
Transform mage::SceneNode::m_transform  [private]
```

The transform of this scene node.

5.42 mage::SceneNodeVisitor Class Reference

```
#include <scene_node_visitor.hpp>
```

Public Member Functions

- virtual ~SceneNodeVisitor ()
- virtual void VisitCameraNode (CameraNode &camera_node)
- virtual void VisitCameraNode (const CameraNode &camera_node)

Protected Member Functions

- SceneNodeVisitor ()

5.42.1 Detailed Description

A class of scene node visitors.

5.42.2 Constructor & Destructor Documentation

5.42.2.1 ∼SceneNodeVisitor()

```
virtual mage::SceneNodeVisitor::∼SceneNodeVisitor ( ) [virtual]
```

Destructs this scene node visitor.

5.42.2.2 SceneNodeVisitor()

```
mage::SceneNodeVisitor::SceneNodeVisitor ( ) [protected]
```

Constructs a scene node visitor.

5.42.3 Member Function Documentation

5.42.3.1 VisitCameraNode() [1/2]

```
virtual void mage::SceneNodeVisitor::VisitCameraNode (
            CameraNode & camera_node ) [virtual]
```

Visits the given camera node.

Parameters

in	*camera_node*	The camera node.

5.42.3.2 VisitCameraNode() [2/2]

```
virtual void mage::SceneNodeVisitor::VisitCameraNode (
            const CameraNode & camera_node ) [virtual]
```

Visits the given camera node.

Parameters

in	*camera_node*	The camera node.

5.43 mage::Semaphore Class Reference

```
#include <lock.hpp>
```

Public Member Functions

- Semaphore ()
- ∼Semaphore ()
- void Post (uint32_t count=1)
- void Wait ()
- bool TryWait ()

Private Attributes

- HANDLE m_handle

5.43.1 Detailed Description

A class of semaphores.

5.43.2 Constructor & Destructor Documentation

5.43.2.1 Semaphore()

```
mage::Semaphore::Semaphore ( )
```

Constructs a semaphore.

5.43.2.2 ∼Semaphore()

```
mage::Semaphore::∼Semaphore ( )
```

Destructs this semaphore.

5.43.3 Member Function Documentation

5.43.3.1 Post()

```
void mage::Semaphore::Post (
             uint32_t count = 1 )
```

Increments the value of this semaphore variable by the given value.

The process executing wait is blocked until the value of the semaphore is greater or equal to 1.

Parameters

`in`	*count*	The increment value.

5.43.3.2 TryWait()

```
bool mage::Semaphore::TryWait ( )
```

Checks whether waiting for this semaphore would be necessary.

Returns

`true` if waiting for this semaphore would be necessary. `false` otherwise.

5.43.3.3 Wait()

```
void mage::Semaphore::Wait ( )
```

Decrements the value of this semaphore variable by one.

If the initial value of the semaphore is negative, the waiting queue is not empty and thus one blocked process can be transferred to the ready queue.

5.43.4 Member Data Documentation

5.43.4.1 m_handle

```
HANDLE mage::Semaphore::m_handle [private]
```

The handle of this semaphore.

5.44 mage::State Class Reference

```
#include <state.hpp>
```

Public Member Functions

- State (uint64_t id=0)
- ∼State ()
- uint64_t GetId () const
- virtual void RequestViewSetup (ViewerSetup ∗viewer_setup)

Protected Member Functions

- virtual void Load ()
- virtual void Close ()
- virtual void Update (double elapsed_time)
- virtual void Render ()

Private Attributes

- const uint64_t m_id

Friends

- class StateManager

5.44.1 Detailed Description

A class of states

5.44.2 Constructor & Destructor Documentation

5.44.2.1 State()

```
mage::State::State (
              uint64_t id = 0 )
```

Constructs a state with given id.

Parameters

in	*id*	The id.

5.44.2.2 ∼State()

```
mage::State::∼State ( )
```

Destructs this state.

(This destructor is not allowed to be called manually: Real destruction must take place in State::Close())

5.44.3 Member Function Documentation

5.44.3.1 Close()

```
virtual void mage::State::Close ( )  [protected], [virtual]
```

Closes this state. Allows this state to preform any post-processing destruction.

5.44.3.2 GetId()

```
uint64_t mage::State::GetId ( ) const
```

Returns the id of this state.

Returns

The id of this state.

5.44.3.3 Load()

```
virtual void mage::State::Load ( )  [protected], [virtual]
```

Loads this state. Allows this state to preform any pre-processing construction.

5.44.3.4 Render()

```
virtual void mage::State::Render ( )  [protected], [virtual]
```

Render this state.

5.44.3.5 RequestViewSetup()

```
virtual void mage::State::RequestViewSetup (
            ViewerSetup * viewer_setup )  [virtual]
```

Requests the view setup details for the given frame.

Precondition

viewer_setup is not `nullptr`.

Parameters

in,out	*viewer_setup*	A pointer to a viewer setup.

5.44.3.6 Update()

```
virtual void mage::State::Update (
            double elapsed_time )  [protected], [virtual]
```

Updates this state.

Parameters

in	*elapsed_time*	The elapsed time since the previous update.

5.44.4 Friends And Related Function Documentation

5.44.4.1 StateManager

```
friend class StateManager  [friend]
```

5.44.5 Member Data Documentation

5.44.5.1 m_id

```
const uint64_t mage::State::m_id  [private]
```

Application defined identifier (must be unique for state switching) of this state.

5.45 mage::StateManager Class Reference

```
#include <state_manager.hpp>
```

Public Member Functions

- void AddState (State ∗state, bool change=true)
- void RemoveState (State ∗state)
- void ChangeState (uint64_t id)
- State ∗ GetCurrentState () const
- bool IsStateChanged () const

Protected Member Functions

- StateManager ()
- virtual ∼StateManager ()
- bool Update (double elapsed_time)
- void ChangeState (State ∗state)

Protected Attributes

- list< State ∗> m_states
- State ∗ m_current_state
- bool m_state_changed

Friends

- class Engine

5.45.1 Detailed Description

A class of state managers.

5.45.2 Constructor & Destructor Documentation

5.45.2.1 StateManager()

```
mage::StateManager::StateManager ( )  [protected]
```

Constructs a state manager.

5.45.2.2 ∼StateManager()

```
virtual mage::StateManager::∼StateManager ( )  [protected], [virtual]
```

Destructs this state manager.

5.45.3 Member Function Documentation

5.45.3.1 AddState()

```
void mage::StateManager::AddState (
            State * state,
            bool change = true )
```

Adds the given state from the states of this state manager.

Parameters

in	*state*	A pointer to the state.
in	*change*	Flag indicating whether the current state of this engine need to be changed to *state*.

5.45.3.2 ChangeState() [1/2]

```
void mage::StateManager::ChangeState (
            uint64_t id )
```

Changes the state of this state manager to the state with the given id.

Parameters

in	*id*	The id of the state to change to.

5.45.3.3 ChangeState() [2/2]

```
void mage::StateManager::ChangeState (
             State * state )  [protected]
```

Changes the state of this state manager to the given state.

Precondition

state is not `nullptr`.

Parameters

in	*state*	A pointer to the new state.

5.45.3.4 GetCurrentState()

```
State* mage::StateManager::GetCurrentState ( ) const
```

Returns the current state of this state manager.

Returns

A pointer to the current state of this state manager.

5.45.3.5 IsStateChanged()

```
bool mage::StateManager::IsStateChanged ( ) const
```

Checks whether the state of this state manager is changed.

Returns

`true` if the state is changed. `false` otherwise.

5.45.3.6 RemoveState()

```
void mage::StateManager::RemoveState (
             State * state )
```

Removes (and destructs) the given state from the states of this state manager.

Parameters

in	*state*	A pointer to the state.

5.45.3.7 Update()

```
bool mage::StateManager::Update (
             double elapsed_time )  [protected]
```

Updates this state manager and its current state.

Parameters

in	*elapsed_time*	The elapsed time since the previous frame.

Returns

`true` if the state is changed in the current frame. `false` otherwise.

5.45.4 Friends And Related Function Documentation

5.45.4.1 Engine

```
friend class Engine  [friend]
```

5.45.5 Member Data Documentation

5.45.5.1 m_current_state

```
State* mage::StateManager::m_current_state  [protected]
```

A pointer to the current state of this state manager.

5.45.5.2 m_state_changed

```
bool mage::StateManager::m_state_changed  [protected]
```

Flag indicating if the state changed in the current frame.

5.45.5.3 m_states

```
list< State * > mage::StateManager::m_states  [protected]
```

The states of this state manager.

5.46 mage::Task Class Reference

```
#include <task.hpp>
```

Public Member Functions

- virtual ∼Task ()
- virtual void Run ()=0

5.46.1 Constructor & Destructor Documentation

5.46.1.1 ∼Task()

```
virtual mage::Task::∼Task ( )  [virtual]
```

5.46.2 Member Function Documentation

5.46.2.1 Run()

```
virtual void mage::Task::Run ( )  [pure virtual]
```

5.47 mage::Timer Class Reference

```
#include <timer.hpp>
```

Public Member Functions

- Timer ()
- virtual ∼Timer ()
- void Start ()
- void Stop ()
- void Reset ()
- void Restart ()
- double Time ()

Protected Member Functions

- double time ()

Protected Attributes

- double m_time0
- double m_elapsed
- bool m_running
- LARGE_INTEGER m_performance_counter
- LARGE_INTEGER m_performance_frequency
- double m_performance_period

5.47.1 Detailed Description

A class of (high precision) timers.

5.47.2 Constructor & Destructor Documentation

5.47.2.1 Timer()

```
mage::Timer::Timer ( )
```

Constructs a timer.

5.47.2.2 ∼Timer()

```
virtual mage::Timer::∼Timer ( )  [virtual]
```

Destructs this timer.

5.47.3 Member Function Documentation

5.47.3.1 Reset()

```
void mage::Timer::Reset ( )
```

Resets this timer.

5.47.3.2 Restart()

```
void mage::Timer::Restart ( )
```

Restarts this timer.

5.47.3.3 Start()

```
void mage::Timer::Start ( )
```

Starts this timer.

5.47.3.4 Stop()

```
void mage::Timer::Stop ( )
```

Stops this timer.

5.47.3.5 Time()

```
double mage::Timer::Time ( )
```

Returns the elapsed time of this timer.

Returns

The elapsed time of this timer.

5.47.3.6 time()

```
double mage::Timer::time ( )  [protected]
```

Returns the time of this timer.

Returns

The time of this timer.

Note

This member method encapsulates the performance of the underlying counter/frequency processing.

5.47.4 Member Data Documentation

5.47.4.1 m_elapsed

```
double mage::Timer::m_elapsed  [protected]
```

The elapsed time of this timer.

5.47.4.2 m_performance_counter

```
LARGE_INTEGER mage::Timer::m_performance_counter  [protected]
```

The counter of this timer.

5.47.4.3 m_performance_frequency

```
LARGE_INTEGER mage::Timer::m_performance_frequency  [protected]
```

The frequency of this timer.

5.47.4.4 m_performance_period

```
double mage::Timer::m_performance_period  [protected]
```

The period of this timer.

5.47.4.5 m_running

```
bool mage::Timer::m_running  [protected]
```

Flag indicating whether this timer is running.

5.47.4.6 m_time0

```
double mage::Timer::m_time0  [protected]
```

The initial time stamp of this timer.

5.48 mage::TLVertex Struct Reference

```
#include <vertex.hpp>
```

Public Member Functions

- TLVertex ()
- TLVertex (XMFLOAT4 p, XMFLOAT4 diffuse, XMFLOAT2 tex)

Public Attributes

- XMFLOAT4 p
- XMFLOAT4 diffuse
- XMFLOAT2 tex

5.48.1 Detailed Description

A struct of transformed and lit vertices.

5.48.2 Constructor & Destructor Documentation

5.48.2.1 TLVertex() [1/2]

```
mage::TLVertex::TLVertex ( )
```

Constructs a transformed and lit vertex.

5.48.2.2 TLVertex() [2/2]

```
mage::TLVertex::TLVertex (
             XMFLOAT4 p,
             XMFLOAT4 diffuse,
             XMFLOAT2 tex )
```

Constructs a transformed and lit vertex.

Parameters

in	*p*	The position of the transformed and lit vertex (in projection space).
in	*diffuse*	The diffuse colour of the transformed and lit vertex.
in	*tex*	The texture coordinates of the transformed and lit vertex.

5.48.3 Member Data Documentation

5.48.3.1 diffuse

```
XMFLOAT4 mage::TLVertex::diffuse
```

The diffuse colour of this transformed and lit vertex.

5.48.3.2 p

```
XMFLOAT4 mage::TLVertex::p
```

The position of this transformed and lit vertex (in projection space).

5.48.3.3 tex

```
XMFLOAT2 mage::TLVertex::tex
```

The texture coordinates of this transformed and lit vertex.

5.49 mage::Transform Struct Reference

```
#include <transform.hpp>
```

Public Member Functions

- Transform (const CartesianAxesSystem &axes)
- Transform (const CartesianCoordinateSystem &coordinate_system)
- Transform (const XMFLOAT3 &translation={ 0.0f, 0.0f, 0.0f }, const XMFLOAT3 &rotation={ 0.0f, 0.0f, 0.0f }, const XMFLOAT3 &scale={ 1.0f, 1.0f, 1.0f })
- Transform (const Transform &transform)
- ∼Transform ()
- Transform & operator= (const Transform &transform)
- Transform & SetTranslationX (float x)
- Transform & SetTranslationY (float y)
- Transform & SetTranslationZ (float z)
- Transform & SetTranslation (float x, float y, float z)
- Transform & SetTranslation (const XMFLOAT3 &translation)
- void AddTranslationX (float x)
- void AddTranslationY (float y)

- void AddTranslationZ (float z)
- void AddTranslation (float x, float y, float z)
- void AddTranslation (const XMFLOAT3 &translation)
- float GetTranslationX () const
- float GetTranslationY () const
- float GetTranslationZ () const
- XMFLOAT3 GetTranslation () const
- XMMATRIX GetTranslationMatrix () const
- Transform & SetRotationX (float x)
- Transform & SetRotationY (float y)
- Transform & SetRotationZ (float z)
- Transform & SetRotation (float x, float y, float z)
- Transform & SetRotation (const XMFLOAT3 &rotation)
- Transform & SetRotationAroundDirection (const XMVECTOR &normal, float angle)
- void AddRotationX (float x)
- void AddRotationY (float y)
- void AddRotationZ (float z)
- void AddRotation (float x, float y, float z)
- void AddRotation (const XMFLOAT3 &rotation)
- float GetRotationX () const
- float GetRotationY () const
- float GetRotationZ () const
- XMFLOAT3 GetRotation () const
- XMMATRIX GetRotationMatrix () const
- Transform & SetScaleX (float x)
- Transform & SetScaleY (float y)
- Transform & SetScaleZ (float z)
- Transform & SetScale (float x, float y, float z)
- Transform & SetScale (const XMFLOAT3 &scale)
- void AddScaleX (float x)
- void AddScaleY (float y)
- void AddScaleZ (float z)
- void AddScale (float x, float y, float z)
- void AddScale (const XMFLOAT3 &scale)
- float GetScaleX () const
- float GetScaleY () const
- float GetScaleZ () const
- XMFLOAT3 GetScale () const
- XMMATRIX GetScaleMatrix () const
- XMVECTOR GetLocalAxisX () const
- XMVECTOR GetLocalAxisY () const
- XMVECTOR GetLocalAxisZ () const
- CartesianAxesSystem GetLocalAxes () const
- CartesianCoordinateSystem GetLocalCoordinateSystem () const
- XMVECTOR GetWorldAxisX () const
- XMVECTOR GetWorldAxisY () const
- XMVECTOR GetWorldAxisZ () const
- CartesianAxesSystem GetWorldAxes () const
- CartesianCoordinateSystem GetWorldCoordinateSystem () const
- XMVECTOR GetLocalLeft () const
- XMVECTOR GetLocalUp () const
- XMVECTOR GetLocalForward () const
- XMVECTOR GetWorldLeft () const
- XMVECTOR GetWorldUp () const
- XMVECTOR GetWorldForward () const
- XMMATRIX GetWorldToObjectMatrix () const
- XMMATRIX GetObjectToWorldMatrix () const
- XMMATRIX GetWorldToViewMatrix () const

Private Member Functions

- XMMATRIX GetInverseTranslationMatrix () const
- XMMATRIX GetInverseRotationMatrix () const
- XMMATRIX GetInverseScaleMatrix () const
- XMVECTOR TransformObjectToWorldDirection (const XMVECTOR &direction) const

Private Attributes

- XMFLOAT3 m_translation
- XMFLOAT3 m_rotation
- XMFLOAT3 m_scale

5.49.1 Detailed Description

A struct of transforms.

5.49.2 Constructor & Destructor Documentation

5.49.2.1 Transform() [1/4]

```
mage::Transform::Transform (
             const CartesianAxesSystem & axes )
```

Constructs a transform from the given local Cartesian axes system.

Parameters

in	*axes*	The local Cartesian axes system.

5.49.2.2 Transform() [2/4]

```
mage::Transform::Transform (
             const CartesianCoordinateSystem & coordinate_system )
```

Constructs a transform from the given local Cartesian coordinate system.

Parameters

in	*coordinate_system*	The local Cartesian coordinate system.

5.49.2.3 Transform() [3/4]

```
mage::Transform::Transform (
             const XMFLOAT3 & translation = { 0.0f, 0.0f, 0.0f },
             const XMFLOAT3 & rotation = { 0.0f, 0.0f, 0.0f },
             const XMFLOAT3 & scale = { 1.0f, 1.0f, 1.0f } )
```

Constructs a transform from the given translation, rotation and scale component.

Parameters

in	*translation*	The translation component.
in	*rotation*	The rotation component.
in	*scale*	The scale component.

5.49.2.4 Transform() [4/4]

```
mage::Transform::Transform (
            const Transform & transform )
```

Constructs a transform from the given transform.

Parameters

in	*transform*	The transform.

5.49.2.5 ∼Transform()

```
mage::Transform::∼Transform ( )
```

Destructs this transform.

5.49.3 Member Function Documentation

5.49.3.1 AddRotation() [1/2]

```
void mage::Transform::AddRotation (
            float x,
            float y,
            float z )
```

Adds the given rotation component to the rotation component of this transform.

Parameters

in	*x*	The x-value of the rotation component to add.
in	*y*	The y-value of the rotation component to add.
in	*z*	The z-value of the rotation component to add.

5.49.3.2 AddRotation() [2/2]

```
void mage::Transform::AddRotation (
             const XMFLOAT3 & rotation )
```

Adds the given rotation component to the rotation component of this transform.

Parameters

in	*rotation*	A reference to the rotation component to add.

5.49.3.3 AddRotationX()

```
void mage::Transform::AddRotationX (
             float x )
```

Adds the given x-value to the rotation component of this transform.

Parameters

in	*x*	The x-value of the rotation component to add.

5.49.3.4 AddRotationY()

```
void mage::Transform::AddRotationY (
             float y )
```

Adds the given y-value to the rotation component of this transform.

Parameters

in	*y*	The y-value of the rotation component to add.

5.49.3.5 AddRotationZ()

```
void mage::Transform::AddRotationZ (
             float z )
```

Adds the given z-value to the rotation component of this transform.

Parameters

in	*z*	The z-value of the rotation component to add.

5.49.3.6 AddScale() [1/2]

```
void mage::Transform::AddScale (
             float x,
             float y,
             float z )
```

Adds the given scale component to the scale component of this transform.

Parameters

in	*x*	The x-value of the scale component to add.
in	*y*	The y-value of the scale component to add.
in	*z*	The z-value of the scale component to add.

5.49.3.7 AddScale() [2/2]

```
void mage::Transform::AddScale (
             const XMFLOAT3 & scale )
```

Adds the given scale component to the scale component of this transform.

Parameters

in	*scale*	A reference to the scale component to add.

5.49.3.8 AddScaleX()

```
void mage::Transform::AddScaleX (
             float x )
```

Adds the given x-value to the scale component of this transform.

Parameters

in	*x*	The x-value of the scale component to add.

5.49.3.9 AddScaleY()

```
void mage::Transform::AddScaleY (
             float y )
```

Adds the given y-value to the scale component of this transform.

Parameters

in	*y*	The y-value of the scale component to add.

5.49.3.10 AddScaleZ()

```
void mage::Transform::AddScaleZ (
            float z )
```

Adds the given z-value to the scale component of this transform.

Parameters

in	*z*	The z-value of the scale component to add.

5.49.3.11 AddTranslation() [1/2]

```
void mage::Transform::AddTranslation (
            float x,
            float y,
            float z )
```

Adds the given translation component to the translation component of this transform.

Parameters

in	*x*	The x-value of the translation component to add.
in	*y*	The y-value of the translation component to add.
in	*z*	The z-value of the translation component to add.

5.49.3.12 AddTranslation() [2/2]

```
void mage::Transform::AddTranslation (
            const XMFLOAT3 & translation )
```

Adds the given translation component to the translation component of this transform.

Parameters

in	*translation*	A reference to the translation component to add.

5.49.3.13 AddTranslationX()

```
void mage::Transform::AddTranslationX (
            float x )
```

Adds the given x-value to the translation component of this transform.

Parameters

in	*x*	The x-value of the translation component to add.

5.49.3.14 AddTranslationY()

```
void mage::Transform::AddTranslationY (
            float y )
```

Adds the given y-value to the translation component of this transform.

Parameters

in	*y*	The y-value of the translation component to add.

5.49.3.15 AddTranslationZ()

```
void mage::Transform::AddTranslationZ (
            float z )
```

Adds the given z-value to the translation component of this transform.

Parameters

in	*z*	The z-value of the translation component to add.

5.49.3.16 GetInverseRotationMatrix()

```
XMMATRIX mage::Transform::GetInverseRotationMatrix ( ) const  [private]
```

Returns the inverse rotation matrix of this transform.

Returns

The inverse rotation matrix of this transform.

5.49.3.17 GetInverseScaleMatrix()

```
XMMATRIX mage::Transform::GetInverseScaleMatrix ( ) const  [private]
```

Returns the inverse scale matrix of this transform.

Returns

The inverse scale matrix of this transform.

5.49.3.18 GetInverseTranslationMatrix()

```
XMMATRIX mage::Transform::GetInverseTranslationMatrix ( ) const  [private]
```

Returns the inverse translation matrix of this transform.

Returns

The inverse translation matrix of this transform.

5.49.3.19 GetLocalAxes()

```
CartesianAxesSystem mage::Transform::GetLocalAxes ( ) const
```

Returns the local Cartesian axes system of this transform in local space coordinates.

Returns

The local Cartesian axes system of this transform expressed in local space coordinates.

5.49.3.20 GetLocalAxisX()

```
XMVECTOR mage::Transform::GetLocalAxisX ( ) const
```

Returns the direction of the local x-axis of this transform expressed in local space coordinates.

Returns

The direction of the local x-axis of this transform expressed in local space coordinates.

5.49.3.21 GetLocalAxisY()

```
XMVECTOR mage::Transform::GetLocalAxisY ( ) const
```

Returns the direction of the local y-axis of this transform expressed in local space coordinates.

Returns

The direction of the local y-axis of this transform expressed in local space coordinates.

5.49.3.22 GetLocalAxisZ()

```
XMVECTOR mage::Transform::GetLocalAxisZ ( ) const
```

Returns the direction of the local z-axis of this transform expressed in local space coordinates.

Returns

The direction of the local z-axis of this transform expressed in local space coordinates.

5.49.3.23 GetLocalCoordinateSystem()

```
CartesianCoordinateSystem mage::Transform::GetLocalCoordinateSystem ( ) const
```

Returns the local Cartesian coordinate system of this transform in local space coordinates.

Returns

The local Cartesian coordinate system of this transform expressed in local space coordinates.

5.49.3.24 GetLocalForward()

```
XMVECTOR mage::Transform::GetLocalForward ( ) const
```

Returns the local forward direction of this transform expressed in local space coordinates.

Returns

The local forward direction of this transform expressed in local space coordinates.

5.49.3.25 GetLocalLeft()

```
XMVECTOR mage::Transform::GetLocalLeft ( ) const
```

Returns the local left direction of this transform expressed in local space coordinates.

Returns

The local left direction of this transform expressed in local space coordinates.

5.49.3.26 GetLocalUp()

```
XMVECTOR mage::Transform::GetLocalUp ( ) const
```

Returns the local up direction of this transform expressed in local space coordinates.

Returns

The local up direction of this transform expressed in local space coordinates.

5.49.3.27 GetObjectToWorldMatrix()

```
XMMATRIX mage::Transform::GetObjectToWorldMatrix ( ) const
```

Returns the object-to-world matrix of this transform.

Returns

The object-to-world matrix of this transform.

5.49.3.28 GetRotation()

```
XMFLOAT3 mage::Transform::GetRotation ( ) const
```

Returns the rotation component of this transform.

Returns

The rotation component of this transform.

5.49.3.29 GetRotationMatrix()

```
XMMATRIX mage::Transform::GetRotationMatrix ( ) const
```

Returns the rotation matrix of this transform.

Returns

The rotation matrix of this transform.

5.49.3.30 GetRotationX()

```
float mage::Transform::GetRotationX ( ) const
```

Returns the x-value of the rotation component of this transform.

Returns

The x-value of the rotation component of this transform.

5.49.3.31 GetRotationY()

```
float mage::Transform::GetRotationY ( ) const
```

Returns the y-value of the rotation component of this transform.

Returns

The y-value of the rotation component of this transform.

5.49.3.32 GetRotationZ()

```
float mage::Transform::GetRotationZ ( ) const
```

Returns the z-value of the rotation component of this transform.

Returns

The z-value of the rotation component of this transform.

5.49.3.33 GetScale()

```
XMFLOAT3 mage::Transform::GetScale ( ) const
```

Returns the scale component of this transform.

Returns

The scale component of this transform.

5.49.3.34 GetScaleMatrix()

```
XMMATRIX mage::Transform::GetScaleMatrix ( ) const
```

Returns the scale matrix of this transform.

Returns

The scale matrix of this transform.

5.49.3.35 GetScaleX()

```
float mage::Transform::GetScaleX ( ) const
```

Returns the x-value of the scale component of this transform.

Returns

The x-value of the scale component of this transform.

5.49.3.36 GetScaleY()

```
float mage::Transform::GetScaleY ( ) const
```

Returns the y-value of the scale component of this transform.

Returns

The y-value of the scale component of this transform.

5.49.3.37 GetScaleZ()

```
float mage::Transform::GetScaleZ ( ) const
```

Returns the z-value of the scale component of this transform.

Returns

The z-value of the scale component of this transform.

5.49.3.38 GetTranslation()

```
XMFLOAT3 mage::Transform::GetTranslation ( ) const
```

Returns the translation component of this transform.

Returns

The translation component of this transform.

5.49.3.39 GetTranslationMatrix()

```
XMMATRIX mage::Transform::GetTranslationMatrix ( ) const
```

Returns the translation matrix of this transform.

Returns

The translation matrix of this transform.

5.49.3.40 GetTranslationX()

```
float mage::Transform::GetTranslationX ( ) const
```

Returns the x-value of the translation component of this transform.

Returns

The x-value of the translation component of this transform.

5.49.3.41 GetTranslationY()

```
float mage::Transform::GetTranslationY ( ) const
```

Returns the y-value of the translation component of this transform.

Returns

The y-value of the translation component of this transform.

5.49.3.42 GetTranslationZ()

```
float mage::Transform::GetTranslationZ ( ) const
```

Returns the z-value of the translation component of this transform.

Returns

The z-value of the translation component of this transform.

5.49.3.43 GetWorldAxes()

```
CartesianAxesSystem mage::Transform::GetWorldAxes ( ) const
```

Returns the local Cartesian axes system of this transform expressed in world space coordinates.

Returns

The local Cartesian axes system of this transform expressed in world space coordinates.

5.49.3.44 GetWorldAxisX()

```
XMVECTOR mage::Transform::GetWorldAxisX ( ) const
```

Returns the direction of the local x-axis of this transform expressed in world space coordinates.

Returns

The direction of the local x-axis of this transform expressed in world space coordinates.

5.49.3.45 GetWorldAxisY()

```
XMVECTOR mage::Transform::GetWorldAxisY ( ) const
```

Returns the direction of the local y-axis of this transform expressed in world space coordinates.

Returns

The direction of the local y-axis of this transform expressed in world space coordinates.

5.49.3.46 GetWorldAxisZ()

```
XMVECTOR mage::Transform::GetWorldAxisZ ( ) const
```

Returns the direction of the local z-axis of this transform expressed in world space coordinates.

Returns

The direction of the local z-axis of this transform expressed in world space coordinates.

5.49.3.47 GetWorldCoordinateSystem()

```
CartesianCoordinateSystem mage::Transform::GetWorldCoordinateSystem ( ) const
```

Returns the local Cartesian coordinate system of this transform in world space coordinates.

Returns

The local Cartesian coordinate system of this transform expressed in world space coordinates.

5.49.3.48 GetWorldForward()

```
XMVECTOR mage::Transform::GetWorldForward ( ) const
```

Returns the local forward direction of this transform expressed in world space coordinates.

Returns

The local forward direction of this transform expressed in world space coordinates.

5.49.3.49 GetWorldLeft()

```
XMVECTOR mage::Transform::GetWorldLeft ( ) const
```

Returns the local left direction of this transform expressed in world space coordinates.

Returns

The local left direction of this transform expressed in world space coordinates.

5.49.3.50 GetWorldToObjectMatrix()

```
XMMATRIX mage::Transform::GetWorldToObjectMatrix ( ) const
```

Returns the world-to-object matrix of this transform.

Returns

The world-to-object matrix of this transform.

5.49.3.51 GetWorldToViewMatrix()

```
XMMATRIX mage::Transform::GetWorldToViewMatrix ( ) const
```

Returns the world-to-view matrix of this transform.

Returns

The world-to-view matrix of this transform.

5.49.3.52 GetWorldUp()

```
XMVECTOR mage::Transform::GetWorldUp ( ) const
```

Returns the local up direction of this transform expressed in world space coordinates.

Returns

The local up direction of this transform expressed in world space coordinates.

5.49.3.53 operator=()

```
Transform& mage::Transform::operator= (
             const Transform & transform )
```

Copies the given transform to this transform.

Parameters

in	*transform*	The transform to copy from.

Returns

A reference to the copy of the given transform (i.e. this transform).

5.49.3.54 SetRotation() [1/2]

```
Transform& mage::Transform::SetRotation (
             float x,
             float y,
             float z )
```

Sets the rotation component of this transform to the given rotation component.

Parameters

in	*x*	The x-value of the rotation component.
in	*y*	The y-value of the rotation component.
in	*z*	The z-value of the rotation component.

Returns

A reference to this transform.

5.49.3.55 SetRotation() [2/2]

```
Transform& mage::Transform::SetRotation (
             const XMFLOAT3 & rotation )
```

Sets the rotation component of this transform to the given rotation component.

Parameters

in	*rotation*	A reference to the rotation component.

Returns

A reference to this transform.

5.49.3.56 SetRotationAroundDirection()

```
Transform& mage::Transform::SetRotationAroundDirection (
             const XMVECTOR & normal,
             float angle )
```

Sets the rotation component to a rotation of the given angle around the given normal.

Parameters

in	*normal*	A reference to the normal.
in	*angle*	The angle.

Returns

A reference to this transform.

5.49.3.57 SetRotationX()

```
Transform& mage::Transform::SetRotationX (
             float x )
```

Sets the x-value of the rotation component of this transform to the given value.

Parameters

in	*x*	The x-value of the rotation component.

Returns

A reference to this transform.

5.49.3.58 SetRotationY()

```
Transform& mage::Transform::SetRotationY (
             float y )
```

Sets the y-value of the rotation component of this transform to the given value.

Parameters

in	*y*	The y-value of the rotation component.

Returns

A reference to this transform.

5.49.3.59 SetRotationZ()

```
Transform& mage::Transform::SetRotationZ (
             float z )
```

Sets the z-value of the rotation component of this transform to the given value.

Parameters

in	*z*	The z-value of the rotation component.

Returns

A reference to this transform.

5.49.3.60 SetScale() [1/2]

```
Transform& mage::Transform::SetScale (
             float x,
             float y,
             float z )
```

Sets the scale component of this transform to the given scale component.

Parameters

in	*x*	The x-value of the scale component.
in	*y*	The y-value of the scale component.
in	*z*	The z-value of the scale component.

Returns

A reference to this transform.

5.49.3.61 SetScale() [2/2]

```
Transform& mage::Transform::SetScale (
             const XMFLOAT3 & scale )
```

Sets the scale component of this transform to the given scale component.

Parameters

in	*scale*	A reference to the scale component.

Returns

A reference to this transform.

5.49.3.62 SetScaleX()

```
Transform& mage::Transform::SetScaleX (
             float x )
```

Sets the x-value of the scale component of this transform to the given value.

Parameters

in	*x*	The x-value of the scale component.

Returns

A reference to this transform.

5.49.3.63 SetScaleY()

```
Transform& mage::Transform::SetScaleY (
             float y )
```

Sets the y-value of the scale component of this transform to the given value.

Parameters

in	*y*	The y-value of the scale component.

Returns

A reference to this transform.

5.49.3.64 SetScaleZ()

```
Transform& mage::Transform::SetScaleZ (
             float z )
```

Sets the z-value of the scale component of this transform to the given value.

Parameters

in	*z*	The z-value of the scale component.

Returns

A reference to this transform.

5.49.3.65 SetTranslation() [1/2]

```
Transform& mage::Transform::SetTranslation (
             float x,
             float y,
             float z )
```

Sets the translation component of this transform to the given translation component.

Parameters

in	*x*	The x-value of the translation component.
in	*y*	The y-value of the translation component.
in	*z*	The z-value of the translation component.

Returns

A reference to this transform.

5.49.3.66 SetTranslation() [2/2]

```
Transform& mage::Transform::SetTranslation (
             const XMFLOAT3 & translation )
```

Sets the translation component of this transform to the given translation component.

Parameters

in	*translation*	A reference to the translation component.

Returns

A reference to this transform.

5.49.3.67 SetTranslationX()

```
Transform& mage::Transform::SetTranslationX (
             float x )
```

Sets the x-value of the translation component of this transform to the given value.

Parameters

in	*x*	The x-value of the translation component.

Returns

A reference to this transform.

5.49.3.68 SetTranslationY()

```
Transform& mage::Transform::SetTranslationY (
             float y )
```

Sets the y-value of the translation component of this transform to the given value.

Parameters

`in`	*y*	The y-value of the translation component.

Returns

A reference to this transform.

5.49.3.69 SetTranslationZ()

```
Transform& mage::Transform::SetTranslationZ (
             float z )
```

Sets the z-value of the translation component of this transform to the given value.

Parameters

`in`	*z*	The z-value of the translation component.

Returns

A reference to this transform.

5.49.3.70 TransformObjectToWorldDirection()

```
XMVECTOR mage::Transform::TransformObjectToWorldDirection (
             const XMVECTOR & direction ) const  [private]
```

Transforms the given direction expressed in the local coordinate space of this transform to world coordinate space.

Parameters

`in`	*direction*	A reference to the direction expressed in the local coordinate space of this transform.

Returns

The transformed (normalized) direction expressed in world coordinate space.

5.49.4 Member Data Documentation

5.49.4.1 m_rotation

```
XMFLOAT3 mage::Transform::m_rotation  [private]
```

The rotation component (in radians) of this transform.

5.49.4.2 m_scale

```
XMFLOAT3 mage::Transform::m_scale  [private]
```

The scale component of this transform.

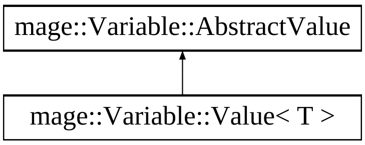
5.49.4.3 m_translation

```
XMFLOAT3 mage::Transform::m_translation  [private]
```

The translation component of this transform.

5.50 mage::Variable::Value< T > Struct Template Reference

Inheritance diagram for mage::Variable::Value< T >:



Public Member Functions

- Value (const T ∗value)
- virtual ∼Value ()
- virtual const void ∗ GetValue () const override

Private Attributes

- const T ∗ m_value

5.50.1 Detailed Description

template<typename T>
struct mage::Variable::Value< T >

A struct of immutable values.

Template Parameters

T	The type of the value.

5.50.2 Constructor & Destructor Documentation

5.50.2.1 Value()

```
template<typename T >
mage::Variable::Value< T >::Value (
             const T * value )
```

Constructs a value.

Parameters

in	*value*	A pointer to the value.

5.50.2.2 ∼Value()

```
template<typename T >
virtual mage::Variable::Value< T >::∼Value ( )  [virtual]
```

Destructs this value.

5.50.3 Member Function Documentation

5.50.3.1 GetValue()

```
template<typename T >
virtual const void* mage::Variable::Value< T >::GetValue ( ) const  [override], [virtual]
```

Returns the value of this value.

Returns

A pointer to the value of this value.

Implements mage::Variable::AbstractValue.

5.50.4 Member Data Documentation

5.50.4.1 m_value

```
template<typename T >
const T* mage::Variable::Value< T >::m_value  [private]
```

A pointer to the value of this value.

5.51 mage::Variable Struct Reference

```
#include <variable.hpp>
```

Classes

- struct AbstractValue
- struct Value

Public Member Functions

- template<typename T >
  Variable (const string &name, VariableType type, const T ∗value)
- ∼Variable ()
- bool operator== (const Variable &variable) const
- bool operator!= (const Variable &variable) const
- const string & GetName () const
- const VariableType & GetType () const
- const void ∗ GetValue () const
- template<typename T >
  void SetValue (const T ∗value)

Private Attributes

- const string m_name
- const VariableType m_type
- const AbstractValue ∗ m_value

5.51.1 Detailed Description

A struct of (immutable) variables.

5.51.2 Constructor & Destructor Documentation

5.51.2.1 Variable()

```
template<typename T >
mage::Variable::Variable (
            const string & name,
            VariableType type,
            const T * value )
```

Constructs a variable.

Template Parameters

T	The (storage) type of the value.

Parameters

in	*name*	The name.
in	*type*	The (scripting) type of the value.
in	*value*	A pointer to the value.

5.51.2.2 ∼Variable()

```
mage::Variable::∼Variable ( )
```

Destructs this variable.

5.51.3 Member Function Documentation

5.51.3.1 GetName()

```
const string& mage::Variable::GetName ( ) const
```

Returns the name of this variable.

Returns

A reference to the name of this variable.

5.51.3.2 GetType()

```
const VariableType& mage::Variable::GetType ( ) const
```

Returns the scripting type of this value.

Returns

The type of this value.

5.51.3.3 GetValue()

```
const void* mage::Variable::GetValue ( ) const
```

Returns the value of this variable.

Returns

A pointer to the value of this variable.

5.51.3.4 operator"!=()

```
bool mage::Variable::operator!= (
             const Variable & variable ) const
```

Checks whether the given variable is not equal to this variable.

Parameters

in	*variable*	A reference to the variable to compare with.

Returns

`true` if and only if this variable and *variable* have not the same name. `false` otherwise.

5.51.3.5 operator==()

```
bool mage::Variable::operator== (
             const Variable & variable ) const
```

Checks whether the given variable is equal to this variable.

Parameters

in	*variable*	A reference to the variable to compare with.

Returns

`true` if and only if this variable and *variable* have the same name. `false` otherwise.

5.51.3.6 SetValue()

```
template<typename T >
void mage::Variable::SetValue (
             const T * value )
```

Sets the value of this variable.

Template Parameters

T	The (storage) type of the value.

Parameters

in	*value*	A pointer to the value.

5.51.4 Member Data Documentation

5.51.4.1 m_name

```
const string mage::Variable::m_name  [private]
```

The name of this variable.

5.51.4.2 m_type

```
const VariableType mage::Variable::m_type  [private]
```

The type of this value.

Note

It is not possible to use typeid(T).name() since this assumes a bijection between the scripting types and the storage types, which is not the case. Thus the type needs to be stored explicitly.

5.51.4.3 m_value

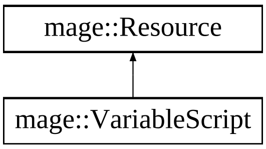
```
const AbstractValue* mage::Variable::m_value  [private]
```

A pointer to the value of this variable.

5.52 mage::VariableScript Class Reference

```
#include <variable_script.hpp>
```

Inheritance diagram for mage::VariableScript:



Public Member Functions

- VariableScript (const string &name, const string &path="./")
- virtual ∼VariableScript ()
- void ImportScript (const string &filename="")
- void ExportScript (const string &filename="")
- bool IsEmpty () const
- size_t GetNbOfVariables () const
- template<typename T >
  void AddVariable (const string &name, VariableType type, const T ∗value)
- void RemoveVariable (const string &name)
- template<typename T >
  const T ∗ GetValueOfVariable (const string &name) const
- template<typename T >
  void SetValueOfVariable (const string &name, const T ∗value)

Protected Member Functions

- void ImportVariable (const string &name, FILE ∗file)
- void ExportVariable (const Variable ∗variable, FILE ∗file)

Protected Attributes

- list< Variable ∗> m_variables

5.52.1 Detailed Description

A class of variable scripts.

5.52.2 Constructor & Destructor Documentation

5.52.2.1 VariableScript()

```
mage::VariableScript::VariableScript (
             const string & name,
             const string & path = "./" )
```

Constructs a variable script.

Parameters

in	*name*	A reference to the name of the variable script.
in	*path*	A reference to the path of the variable script.

5.52.2.2 ∼VariableScript()

```
virtual mage::VariableScript::∼VariableScript ( )  [virtual]
```

Destruct this variable script.

5.52.3 Member Function Documentation

5.52.3.1 AddVariable()

```
template<typename T >
void mage::VariableScript::AddVariable (
             const string & name,
             VariableType type,
             const T ∗ value )
```

Adds the given variable to this variable script.

Precondition

No variable with the name *name* exists in this variable script.

Template Parameters

T	The type of the value.

Parameters

in	*name*	The name of the variable.
in	*type*	The type of the variable.
in	*value*	A pointer to the value of the variable.

5.52.3.2 ExportScript()

```
void mage::VariableScript::ExportScript (
            const string & filename = "" )
```

Exports this variable script to the file with the given filename.

Parameters

in	*filename*	A reference to the filename.

5.52.3.3 ExportVariable()

```
void mage::VariableScript::ExportVariable (
            const Variable * variable,
            FILE * file ) [protected]
```

Export the given variable from this variable script to the given file.

Parameters

in	*variable*	A pointer to the variable variable.
in	*file*	A pointer to a file used for exporting.

5.52.3.4 GetNbOfVariables()

```
size_t mage::VariableScript::GetNbOfVariables ( ) const
```

Returns the number of variables in this variable script.

Returns

The number of variables in this variable script.

5.52.3.5 GetValueOfVariable()

```
template<typename T >
const T* mage::VariableScript::GetValueOfVariable (
             const string & name ) const
```

Returns the value of the given variable in this variable script.

Template Parameters

T	The type of the value.

Parameters

in	*name*	The name of the variable.

Returns

`nullptr` if no variable with the name *name* exists in this variable script.
A pointer to the value of the variable.

5.52.3.6 ImportScript()

```
void mage::VariableScript::ImportScript (
             const string & filename = "" )
```

Imports this variable script from its associated file.

Parameters

in	*filename*	A reference to the filename.

5.52.3.7 ImportVariable()

```
void mage::VariableScript::ImportVariable (
             const string & name,
             FILE * file )  [protected]
```

Import the given variable from the given file to this variable script.

Precondition

No variable with the name *name* exists in this variable script.

Parameters

in	*name*	The name of the variable.
in	*file*	A pointer to a file used for importing.

5.52.3.8 IsEmpty()

```
bool mage::VariableScript::IsEmpty ( ) const
```

Checks wether this variable script is empty.

Returns

`true` if this variable script is empty. `false` otherwise.

5.52.3.9 RemoveVariable()

```
void mage::VariableScript::RemoveVariable (
            const string & name )
```

Removes (and destructs) the given variable from this variable script.

Parameters

in	*name*	The name of the variable.

5.52.3.10 SetValueOfVariable()

```
template<typename T >
void mage::VariableScript::SetValueOfVariable (
            const string & name,
            const T * value )
```

Sets the value of the given variable in this variable script.

Template Parameters

T	The type of the value.

Parameters

in	*name*	The name of the variable.
in	*value*	A pointer to the value of the variable.

Note

Nothing happens if no variable with the name *name* exists in this variable script.

5.52.4 Member Data Documentation

5.52.4.1 m_variables

```
list< Variable * > mage::VariableScript::m_variables [protected]
```

Linked list containing the variables in this variable script.

5.53 mage::Vertex Struct Reference

```
#include <vertex.hpp>
```

Public Member Functions

- Vertex ()
- Vertex (XMFLOAT3 p, XMFLOAT3 n, XMFLOAT2 tex)

Public Attributes

- XMFLOAT3 p
- XMFLOAT3 n
- XMFLOAT2 tex

5.53.1 Detailed Description

A struct of vertices.

5.53.2 Constructor & Destructor Documentation

5.53.2.1 Vertex() [1/2]

```
mage::Vertex::Vertex ( )
```

Constructs a vertex.

5.53.2.2 Vertex() [2/2]

```
mage::Vertex::Vertex (
            XMFLOAT3 p,
            XMFLOAT3 n,
            XMFLOAT2 tex )
```

Constructs a vertex.

Precondition

The length (L2-norm) of the normal must be equal to one (i.e. the normal vector is normalized).

Parameters

in	*p*	The position of the vertex (in object space).
in	*n*	The normal of the vertex.
in	*tex*	The texture coordinates of the vertex.

5.53.3 Member Data Documentation

5.53.3.1 n

```
XMFLOAT3 mage::Vertex::n
```

The normal of this vertex.

5.53.3.2 p

```
XMFLOAT3 mage::Vertex::p
```

The position of this vertex (in object space).

5.53.3.3 tex

```
XMFLOAT2 mage::Vertex::tex
```

The texture coordinates of this vertex.

5.54 mage::ViewerSetup Struct Reference

```
#include <state.hpp>
```

Public Member Functions

- ViewerSetup ()

Public Attributes

- uint64_t m_view_clear_flags

5.54.1 Detailed Description

A struct of viewer setups.

5.54.2 Constructor & Destructor Documentation

5.54.2.1 ViewerSetup()

```
mage::ViewerSetup::ViewerSetup ( )
```

Constructs a viewer setup.

5.54.3 Member Data Documentation

5.54.3.1 m_view_clear_flags

```
uint64_t mage::ViewerSetup::m_view_clear_flags
```

Flags used for clearing the view.