SENTINEL

MANUAL

Table of Contents

[Naming Conventions 3](#_Toc364505838)

[C++ / CLR Wrappers and References 3](#_Toc364505839)

[Create Custom Program 3](#_Toc364505840)

[Class Reference 4](#_Toc364505841)

[Macros 4](#_Toc364505842)

[Global Functions 4](#_Toc364505843)

# Naming Conventions

All class variables start with m, e.g. ColorRGBA\* mRef;

All defines/macros and static variables are in all capitals, e.g. double DESIRED\_FRAME\_RATE;

All member functions start with capital letters, e.g. GameWorld::Inst()->Startup();

All local variables start with lowercase letters, e.g. MeshBuilder meshBuilder;

# C++ / CLR Wrappers and References

Classes named with W represent Wrapper classes. They each possess an instantiation of a variable, and free automatically through the Finalizer. Call Dispose to remove the internal variable from memory before C# does its garbage collection.

Components are the exception to the Wrapper classes in that they create a new GameComponent, but they do not free the memory automatically, except through their associated GameObject. Delete must be called to free the memory if Removing a GameComponent from the GameObject. This behavior also occurs with GameWorld and their GameObject(s).

Classes named with R represent Reference classes. These classes reference a variable. Calling Dispose has no effect on the variable as it is located elsewhere in memory.

See class descriptions for further details relating to shared\_ptr objects.

# Create Custom Program

Use “Sentinel\_Test.cpp” as a reference.

# Class Reference

## Macros

#define TRACE( text )

Outputs text to console.

Example:

TRACE( "Output to Console Successful!" );

## Global Functions

void SetDirectory( const char\* dest );

Sets the directory in relation to the current directory.

Example:

SetDirectory( "Shaders" );

// Load Shaders.

SetDirectory( ".." );