**Game Engine Document**

CODE OR DIE

Austin Eathorne 100 843 298

Neil Hansen 100 897 307

**Team Agreement Document**

**Style Guide**

Overview

There are a few core guidelines that we will be following for our engine. The goal of our guidelines are simple, to provide a consistent and fluent code base that is easily maintainable and readable.

Our guide is based off the C++ Programming Style Guidelines by Geotechnical Software Services: <http://geosoft.no/development/cppstyle.html> , as well as the Google C++ Style Guide: <https://google.github.io/styleguide/cppguide.html#Comment_Style>

For the most part we will be following standard C++ conventions with a few minor preferences added.

Naming Conventions

* Camel Case
  + Type names start with an uppercase letter “ExampleType”
  + Variable names start with a lowercase letter “exampleType”
* Variables
  + Use the prefix n to represent a number of objects “nObjects”
  + Use the prefix “is” should be used for booleans “isInitialized”
  + Variables representing GUI componenets should be suffixed with their component type name “loginText, loginButton”
  + Use names “i, j, k” for iterator variables
* Constants & Enumeration Values
  + All uppercase with words separated by an underscore “DAY\_MONDAY”
* Methods & Functions
  + Must be written using verbs, starting with an uppercase letter “GetName()”
  + Avoid abbreviations “ComputeResult()”, not “CompResult()”
  + Should be named after what they return or what they do in the case of void
* NameSpaces
  + Use all lowercase “common::math::geometry”
* Template Types
  + Use a singular uppercase letter
* Abbreviations & Acronyms
  + Must not be all uppercase when used as a name “GetMp3File()”
* Generic Variables
  + Use the same name as their type “SetAudioClip(AudioClip audioClip)”
* Object Method Name
  + Name of the object is implicit, avoid in method name “object.GetName()”, not “object.GetObjectName()”
* Collections (Arrays & Lists)
  + Variable names representing collections should be plural
* Complement Names
  + Use complimentary names for operations “Start/Stop”

Commenting

* Every class declaration should have a comment describing what the class does
* Use either // or /\* \*/ syntax
* Start comment one space after //, starting with a capital letter
* Do not duplicate comments in the .hpp/.cpp

File

* In general, file naming should be fairly specific
* Filenames should be all lowercase and can include underscores

Code Formatting

* File content must stay within 80 columns
* Split lines must be made obvious (Break after a comma, after an operator, and align with the new beginning of the expression on the previous line)

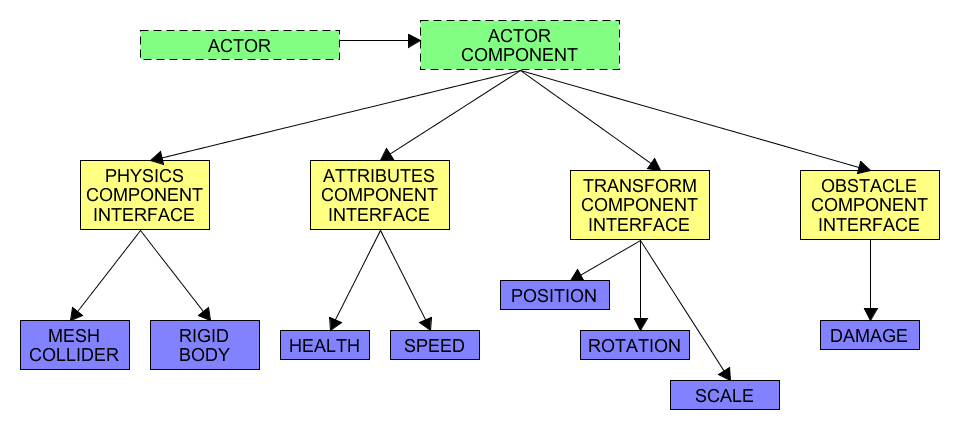
**Project Directory Structure**

* Root
  + Documentation
    - Contains documentation
  + Game
    - Contains executables
  + Source
    - Contains the project files and source code
  + Temp
    - For temporary files

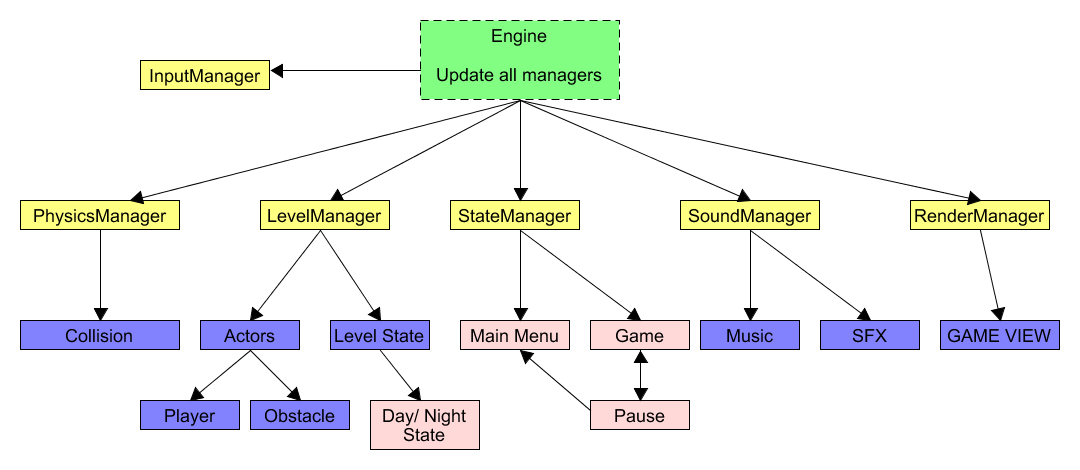
**Minimum PC Requirements**

* Operating System
  + Windows 7
* Processor
  + 2.0 GHz dual-core or better
* Memory
  + 2 GB RAM
* Graphics
  + Nvidia GTX 660/Radeon HD 7800
* Storage
  + 1 GB available space

**Component Architecture Document**

****

**UML Engine Class Structure**

****

**Main Loop Description**

* Update All Managers
  + Input Manager
    - Get User Input
  + State Manager
    - Update State
  + Level Manager
    - Update Level, Actors
  + Collision Manager
    - Check for collision between actors
  + Sound Manager
    - Play Music/Sfx
  + Render Manager
    - Render all visual elements