Lab 03 World Editor Begin

**Fufillment**

* I finished all the objectives!

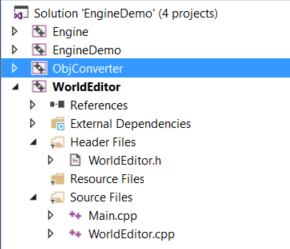
**Execution**

* I do not believe there should be any unexpected requirements for running

**Controls**

* Pressing the X key will close the application
* Pressing W will move the camera forward
* Pressing S will move the camera backward
* Pressing A will move the camera left
* Pressing D will move the camera right
* Holding right click and moving the mouse will turn the camera

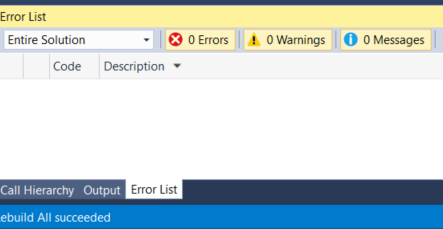
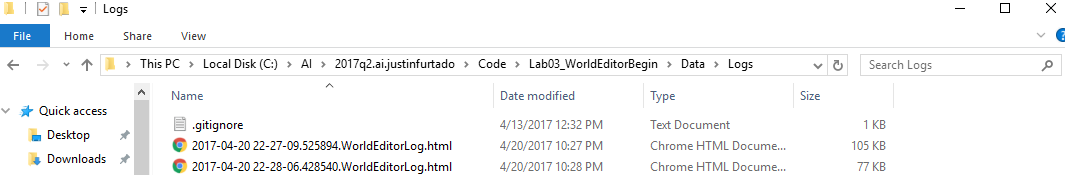
**Screenshots**

* **World Editor**
  + **“show a screenshot of your WorldEditor project in Visual Studio.”**
  + 
  + **“Show your main.cpp (text, not screenshot) as I did above.”**

|  |
| --- |
| **WorldEditor main.cpp** |
| // Order matters  #include "WorldEditor.h"  #include "MyWindow.h"  #pragma warning(push)  #pragma warning(disable : 4127)  #include "QT\qapplication.h"  #pragma warning(pop)  #include "GameLogger.h"  #include "ConfigReader.h"  #include "MyGL.h"  #include <ctime>  int Run(int argc, char \*\*argv)  {  srand((unsigned)time(0));  QApplication app(argc, argv);  WorldEditor game;  Engine::MyWindow \*window = new Engine::MyWindow(&app, &game, WorldEditor::InitializeCallback, WorldEditor::ResizeCallback, WorldEditor::MouseMoveCallback, WorldEditor::DrawCallback, WorldEditor::UpdateCallback, WorldEditor::MouseScrollCallback);  bool fullscreen = false;  Engine::ConfigReader::pReader->GetBoolForKey("WorldEditor.ShowFullscreen", fullscreen);  if (fullscreen)  {  window->showFullScreen();  }  else  {  window->show();  }  int appResult = app.exec();  delete window;  return appResult;  }  const int EXIT\_CONFIG\_FAIL\_INIT = 3;  const int EXIT\_CONFIG\_FAIL\_SHUTDOWN = -3;  int RunWithConfig(int argc, char \*\*argv)  {  Engine::ConfigReader reader;  if (!reader.Initialize("..\\Data\\WorldEditor.config")) return EXIT\_CONFIG\_FAIL\_INIT;  int result = Run(argc, argv);  if (!reader.ShutDown()) return EXIT\_CONFIG\_FAIL\_SHUTDOWN;  return result;  }  const int EXIT\_LOGGER\_FAIL\_INIT = 2;  const int EXIT\_LOGGER\_FAIL\_SHUTDOWN = -2;  int RunWithLogger(int argc, char \*\*argv)  {  if (!Engine::GameLogger::Initialize("..\\Data\\Logs", "WorldEditorLog.html")) return EXIT\_LOGGER\_FAIL\_INIT;  int result = RunWithConfig(argc, argv);  if (!Engine::GameLogger::ShutDown()) return EXIT\_LOGGER\_FAIL\_SHUTDOWN;  return result;  }  int main(int argc, char \*\*argv)  {  int result = RunWithLogger(argc, argv);  system("pause");  return result;  } |

* + **“Same for WorldEditor.config.”**

|  |
| --- |
| **WorldEditor.config** |
| WorldEditor.ShowFullscreen false  WorldEditor.FpsInterval 1.0  EngineDemo.BufferSize 4096  WorldEditor.CameraSpeed 50.0  WorldEditor.CameraRotationSpeed 1.0 |

* + **“Show a screenshot of Daniel’s Hideout.”**
  + 
  + **“Show your build output (without warnings!).”**
  + 
  + 
  + **"**S**how a screenshot of your log folder”**
  + 
    - **Note: I cleared out a few dozen log files before I realized you were looking for this screenshot**

**Post-Mortem**

* Project settings went smoothly, I ran into an issue when making the WorldEditor class that took me over an hour to find, something to do with include files and the compiler telling me that there was a syntax error in a header file referenced by one project but not in that same header file used in other projects, it was quite the strange bug and changing the order of my #includes fixed it.
* I tried to be very thorough when getting rid of unnecessary files and I made sure to copy the bare minimum over from my other project
* I spent 2-3 hours on this lab, counting the error and trying to be thorough in my clearing out of files