Assignment 0: Set-up and Warm-up

CS6323.001 2019 Fall

Total of Points of the Assignment: 3

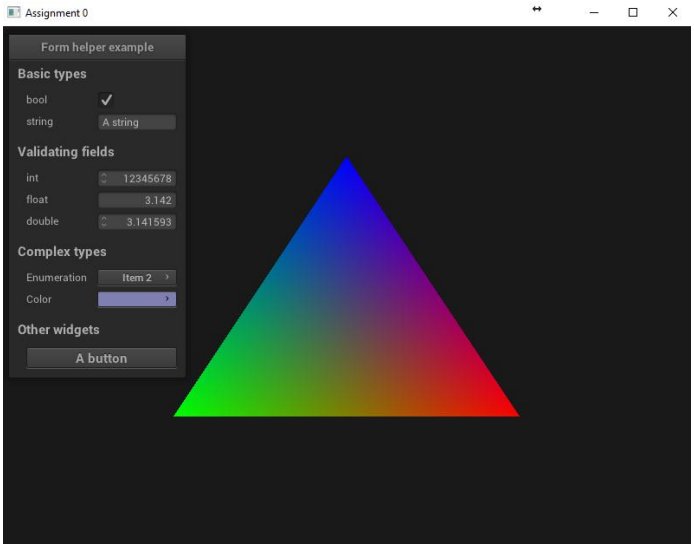
Follow the instructions carefully. If you encounter any problems in the setup, please do not hesitate to come to office hours.

Requirement Overview

This section lists some basic environments/libraries needed in our following class. You are required to set up the OpenGL environment and run a “hello world” program in assignment 0. In the next section, we will go through details in Windows, Mac OS, and Linux.

- C++11
- Graphic card support OpenGL 3.3+
- CMake 2.8+ (If you want to compile the libraries on your own)
- Glew
- Glfw/glut/freeglut
- Glm
- NanoGUI/AntTweakBar
- stb_image

In this assignment, you are required to configure OpenGL shader environment with GUI. The sample code provided NanoGUI as GUI tool.
After finishing this assignment, you are required to demo your code in your computer and the output of your code should be shown like this:



Download the code:

This assignment includes sample code, which can be used to test GLEW, GLFW and GLM configuration, NanoGUI configuration and shader language compilation.

Configuration Tutorial

I introduced basic GLEW, GLFW and GLM compile tutorial in this section. For GUI you can feel free to choose any GUI libs such as NanoGUI/ANtTweakBar and for loading images you can feel free to choose any image libs such as SOIL/FreeImage.

I wrapped binary files for windows VS2015 x64 version(which should be compatible for vs2017 as well): link (https://github.com/kingofyeti/opengl_configuration), includes GLFW, GLEW, GLM, Freeglut, AntTweakBar, NanoGUI and SOIL. It also includes a tutorial link (http://jody-lu-blog.logdown.com/posts/1533525) for configuring MAC OS version of NanoGUI.

Windows (Visual Studio):

1: Download compiled binary file

- GLEW: link (https://sourceforge.net/projects/glew/files/glew/2.0.0/glew-2.0.0-win32.zip/download)
- GLFW: link (https://github.com/glfw/glfw/releases/download/3.2.1/glfw-3.2.1.bin.WIN64.zip)
- GLM: link (https://github.com/g-truc/glm/archive/0.9.8.5.zip)

2: Wrap into folder and add to environment path:

Add this folder into your system variables with:

Variable name: OPENGL

Variable value: YOUR_CURRENT_FOLDER

Add dll files into directly into project folder (Not recommended)

Put all dll files into a folder and add it into “Path” (Recommended)

Variable name: Path

Variable value: YOUR_DLL_FOLDER

3: Open Visual Studio and create an empty project. Add helloworld.cpp file into it

4: Add external libs path and include path in Property pages (or import prop page)

5: Run code

MAC OS (XCode):

1: Homebrew:

- $ brew update
- $ brew install glfw3 glew
- GLM you can directly add it to include header path

2: XCode:

- Add /usr/local/include in header path

- Add /usr/local/glew/2.0.0/lib /usr/local/Cellar/glfw/3.2.1/lib in library path

3: Add link binary libraries

- libglfw.3.2.dylib libGLEW.2.0.0.dylib OpenGL framework

(If you are facing the “GLFW Fails to Open Window in OSX error”(This is one error I face during compilation), add glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);) (http://stackoverflow.com/questions/22887922/glfw-fails-to-open-window-in-osx)

4: Run code

Linux (ubuntu):

1: Install using apt-get

- $ sudo apt-get install libglew-dev libglfw3-dev libglm-dev

2: Compile

- $ sudo g++ -o helloworld helloworld.cpp -std=c++11 -lGL -lglfw -lGLEW

3: Run code