

Lab #3 – cpe471 Fall 2016

Introduction to glfw, OpenGL and GLSL – hello triangle(s) and pixel to World mapping

Due at the end of 2 full lab periods

Today we will alter an existing program in order to get started working with OpenGL, glfw and GLSL. Your goal is to be able to add two new triangles to the scene and to make one vertex of one triangle be repositioned based on a mouse click.

Task 1 - setup:

Your first task for today is to make sure you can compile and run a glfw OpenGL program. All programs from this class from now on must be run locally on the machine you are physically working on. Take the time now to configure your work environment to work locally. Note that in general in order to run this program you will need to have:

the latest graphics driver for your computer (note you should not need to download OpenGL, it came with your computer, but having the latest graphics driver will guarantee you are using the most updated version of OGL).

glfw installed

glew installed

glm installed

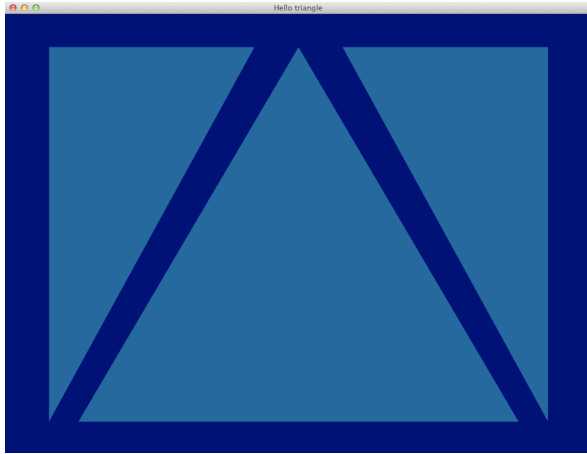
CMake

Please start by downloading the base code from the webpage and reading the readme. It includes specific information about downloading/installing/compiling libraries and setting up your environment variables to point to the appropriate library. Please carefully follow these directions as we will be grading based on the given base code and will expect a consistent set up.

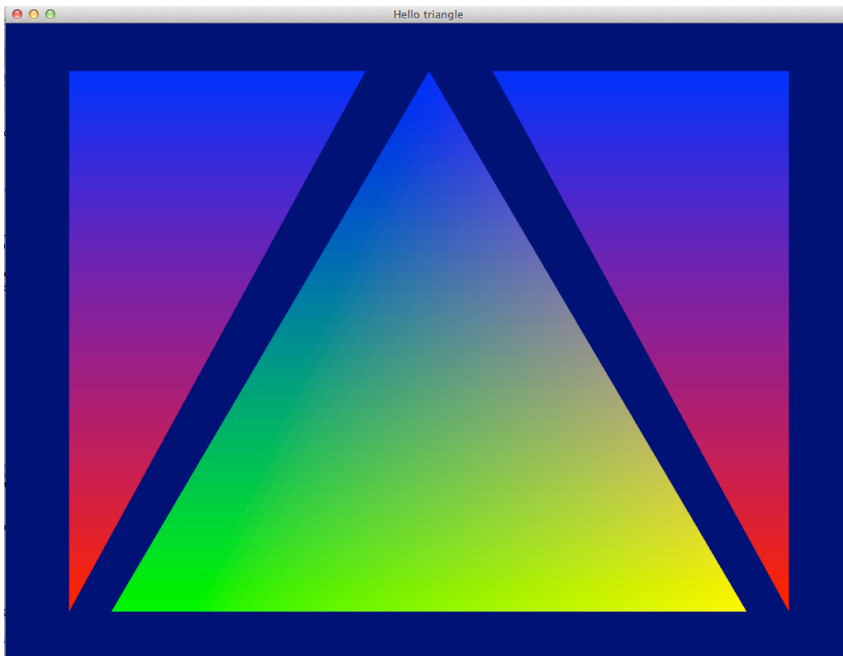
Once you have everything installed and configured, run the code and observe what it does.

Task 2 – add more triangles and colors:

1) Modify the code to add two other triangles to the scene; you will do this by modifying the .cpp file only. Then modify the fragment shader to draw blue triangles, so it should look as follows:



Further modify the code to pass in a different color per vertex. You will need to modify both the .cpp file, the vertex shader and the fragment shader. Your final demo for the lab should look like this:



Task 3 – allow the top vertex of the central triangle to be moved based on mouse input:

1) Modify the code first by writing two helper functions:

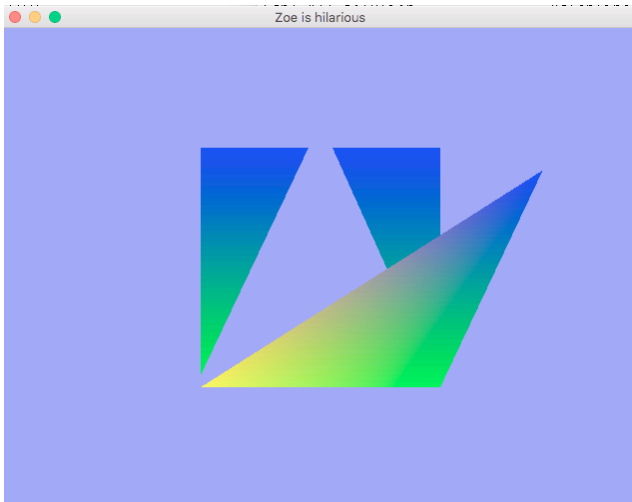
float p2wX(int xp)

float p2wY(int yp)

These functions should:

- take in a coordinate from the mouse,
- convert the Y coordinate to be a standard pixel Y coordinate
- convert both the X and Y into world coordinates

2) Modify the code such that when the mouse is pressed in the screen, the top vertex draws wherever the mouse was pressed – for example (updated coloring for fun):



Demo:

Demo your colored triangles and mouse press movement to the instructor or TA for lab credit. Also be sure to show your p2wx and p2wy code.

References:

As a reference you can take a look at these tutorials for some good introductory information about the libraries we are using (glfw and OpenGL) and some information on what the key syntax in the base code is doing:

<http://www.opengl-tutorial.org/beginners-tutorials/tutorial-1-opening-a-window/>
and

<http://www.opengl-tutorial.org/beginners-tutorials/tutorial-2-the-first-triangle/>