Lab Assignment #2, Part A
Due Oct. 1, 2019

Immediate mode GUI

The goal of this lab is to experiment with immediate mode GUI elements and OpenGL features. We will use the imgui library (https://github.com/ocornut/imgui). Some documentation can be found at the website, some more documentation is in the comments at the top of the *imgui.cpp* file. A good way to learn about imgui is to create a sample program and call `ShowTestWindow()`, then run the program and explore the contents of the test window. It demonstrates the GUI widgets and most of the capabilities of imgui. If you see something you want to copy from the test window you can explore the source code of the `ShowTestWindow` function and grab the code from there.

- Build and run. Get the *GUI demo.zip* file from Blackboard. Build and run the code in Visual Studio.

- Test Window. Explore the contents of the test window. You may need to double-click it to open it. Especially pay attention to the **Widgets** section.

- Imgui intro.
  - Uncomment the referenced line in draw_gui().
  - Find where the angle variable is declared.
  - Find where the angle variable is used.

- Checkboxes.
  - Add a checkbox to enable/disable clearing of the screen. **[5 pts]**
    - Rotate the mesh while clearing is disabled. Can you explain what you see?
  - Add a checkbox to enable/disable depth testing. **[5 pts]**
    - Rotate the mesh while depth testing is disabled. Can you explain what you see?

- Color picker. Add a color picker (ImGui::ColorEdit4) to allow the user to change the color that the screen gets cleared to. **[5 pts]**

- Imgui window.
  - Make the checkbox and color picker you previously created appear in a new imgui window with your name in the imgui window titlebar. **[10 pts]** Hint: ImGui::Begin(…)
  - Add a button to this new window which resets the mesh rotation angle to 0. **[10 pts]**

- Setting uniform variables
  - Create a slider that controls the value of a uniform float variable in the shader. The value should range from 0.0 to 2.0. In the fragment shader, multiply the fragment color by this value to darken/lighten the appearance of the mesh. **[10 pts]**

  - Add another imgui widget which allows you to control some aspect of the rendered mesh. The visual effect is up to you. **[10 pts]**

- Some ideas:

  - Animate the mesh vertices in the vertex shader
  - Animate the texture coordinates in the vertex or fragment shader
  - Apply a pattern to the mesh in the fragment shader