LAB #2

The purpose of this lab is to allow you to become more comfortable with OpenGL and graphics programming, and to organise your base code for your final assignment.

1. This exercise is strictly **individual** (no groupwork).
2. You should **attend the lecture with Donal** on Weds, read the Lab2 slides and watch the Lab2 recording in this folder. To get help from demonstrators, please post your questions on the Discussion Board. There will be no lab session this Thursday, but there will be next week.
3. Create a YouTube video of your program running and submit the link on Blackboard by **Friday October 6th**.
4. Your program should have the following features:

    - Include a maths class (e.g., GLM) and use vectors rather than arrays of floats to represent your vertices. Become comfortable with the Matrix functions and how they work
    - Try to look for 3D objects online to input into your VBOs
    - Try to create multiple VBOs and use multiple shaders to colour them
    - Learn about Uniform Variables and how to pass them into your vertex shader
    - GLUT: Add keyboard and mouse handlers
    - Externalise your shaders: allow your program to input text files for the vertex and fragment shaders
    - Object orientation: Make a Shader class so that you can just create instances of this class each time you need to call a shader
    - Object orientation: Make an object class which sets up buffer objects for your objects.

Notes

- You will need to refer to the documentation of your supporting library e.g. GLFW or FreeGLUT to find the functions and call-backs to use for keyboard input.
- Make sure that you have implemented error checks. Are you checking for shader compilation and linking errors? Do you print the logs in this case? Are you checking that the result of glGetUniformLocation() is not negative?
- Remember to read the OpenGL 4 man-pages for every function that you use from OpenGL - know what the parameters do, and what other functions they depend on.
  The http://docs.gl/project is a very much improved presentation of the man-pages, and has examples.
- Think about the efficiency of your code - what calculations should be done in C, and what should be computed in shaders?