**Assignment 2c: Programming Interactive 3D Graphics with OpenGL and GLFW**

This assignment is designed to provide you with some minimal experience in programming a very basic interactive 3D graphics application using OpenGL and GLFW. The two principal learning objectives are: 1) to help you effectively master the use of viewing transformations to drive a moving camera through a static scene; and 2) to give you experience with managing the definition of an appropriate view frustum and viewport-to-window mapping in the context of a viewing window that changes in size and shape.

Specifically, in this assignment you are asked to:

1) Define a simple 3D model, containing sufficient minimal geometry to allow the viewer to appreciate and understand the effects of different viewing and projection settings. The model can either be hard-coded in your program or read from a file.

2) Define an initial viewing transformation matrix, given the location of the camera, the direction of view (or location of a lookat point, from which the view direction can be inferred), and an 'up' direction that defines the roll of the camera about the view direction. Initial values for these variables can either be hard-coded in your program, or read from a file.

3) Allow the user to re-define the position and orientation of the camera with respect to the surrounding scene using the arrow keys on the keyboard. Pressing the 'up' arrow should move the camera forward in the direction of view; pressing the 'down' arrow should move the camera backwards; pressing the 'left' arrow should rotate the viewing direction to the left (counterclockwise) about the vertical axis through the current camera position; pressing the 'right' arrow should rotate the viewing direction the other way.

4) Define an appropriate perspective transformation matrix. Note that the aspect ratio of the viewing frustum should match the aspect ratio of the window. The perspective transformation matrix can be defined by the locations of the left, right, bottom, top, near and far boundaries of the viewing frustum, given in camera coordinates (relative to the eye), or by the specification of an aspect ratio and one field of view angle (either horizontal or vertical). The initial values can be hard-coded into your program or read from a file. You will want to ensure that the bounds of the viewing frustum are appropriately defined to enclose the full extent of the

scene that you want to be able to view. Please be sure to define your matrix so that only a reasonable (and not unreasonably large) amount of perspective distortion is evident in the image.

5) When the user re-sizes the window, adjust both the dimensions of the viewport and the aspect ratio of the viewing frustum to match the new window dimensions.

Please turn in:

- all of your source code
- one image showing the result of running your program
- a readme file that explains how to compile your program and any special instructions that will be needed to easily build it