CSC 305 Assignment 2 Part 1 - Viewing

Due Monday, March 14, 2015

Minimum Requirements - Part 1 (40% of Assignment 2)

Part 1 of the assignment is for you to practise the viewing transformation and get started with openGL programming. The programming framework will provide you with an "AddLine" method, which will use openGL to draw one straight line segment. The framework will also provide several callback functions for you to fill in. After completing the minimum requirements, you should start to look into the implementation of the framework and try to understand it. This will be very helpful for the second part of assignment 2, which will ask you to do more programming with the raw openGL APIs.

The minimum requirements of part 1 is to implement an interactive wireframe rendering of a cube. The idea is that you write the pipeline of transformations, and use the "AddLine" function of the "Canvas" class transformed by the matrix you compute. You are required to implement the Model, View and Perspective transforms yourself, and applying them to a simple cube.

Features required:

Using only the "AddLine" function provided: 1. Draw a wireframe cube in perspective at the centre of the viewport.
You should write a function to set the camera position and the point it is looking at.

1. Implement camera rotation with the left mouse button: Assuming the camera is at a distance from the cube. When you press the left mouse button down and move the cursor left and right, the camera should rotate around the centre of the object. When you press down the left mouse button and move the cursor up and down, the camera rotates up and down. In both cases the camera should be looking at the centre of the cube or a user specified point.

2. Implement the camera moving along the gaze vector with the right mouse button. When press down the right mouse button and move the cursor up and down, the camera moves closer and further away from the cube, but the viewing direction is not changed. (This is called ‘dolley’ or ‘trolley’.) The provided Canvas class and related call back functions are described later in this document.

A Step Further

After you have completed part 1, please take the Canvas class as an example, try to edit the provided code and make more complicated rendering than wireframes. Look into both the *Canvas* class and the *Image* class we used for assignment one (they're both in the folder "Image" in the framework), and pay attention to the following aspects:

1. Shaders are defined on top of the .cpp file. The compilation of shaders and the way of setting uniform shader variables.
2. The setup of VAO, VBO and the enabling of attribute pointer
3. The draw command, glDrawArrays()
4. Callback function setup with GLFW. This is primarily demonstrated in the Canvas class.

Since you will be demoing both part 1 and part 2 for Assignment 2 marking, we recommend you to save your work in part 1 to a folder, then re-download a copy of the unmodified code framework from ConneX to do further experiments.