ICT289: Computer Graphics Principles and Applications

Semester 1 2021

Practical Labs of Weeks 5 and 6:

Viewing and Projections

The main objectives of these exercises are to:

- Understand the concept of virtual camera
- Understand Viewing and Projections
- Perform viewing and geometric operations using OpenGL
- Extend you C programming skills
- Understand the nature of pointers to functions in C and thus understand the nature of "callbacks"

This exercise is **not** to be submitted for **assessment**.

You should note that even though an exercise is not assessed, not attempting the exercise would make it very difficult for you to understand subsequent material.

The best advice that I can give when doing graphics programming is to start simple – do the exercises in this lab sheet first. These are not the most efficient programs but they are a good starting point for learning, without going off into the deep end.

Exercise 3.1.

The focal length of a camera lens is the distance from the center of the lens to the point at which parallel rays of light will be focused. For a pinhole camera, the focal length is the distance from the pinhole to the film plane. Assuming that the human visual system has an angle of view of 90 degrees, what focal length should we use with 35-mm film to achieve a natural view?

You can assume that the dimensions of a frame of 35-mm film are 24 mm by 36 mm. Check your textbook and lecture notes.

Exercise 3.2.
Write a program to display a wireframe teapot (use glut library function ***glutWireTeapot***). The user should be able to get six different views of the teapot – front, rear, top, bottom, left and right – by pressing keys as follows: **F**(ront), **R**(ear), **T**(op), **B**(ottom), (r)**I**(ght), **L**(eft)

Note: The camera-up vector (the components of which are the last three parameters of gluLookAt) is normally 0.0, 1.0, 0.0, but will need to be different for the bottom and top views. The front view is with the eye at positive z looking at (0, 0, 0).

Exercise 3.3.
Extend the 3D house program that you created in the previous lab. First use orthographic viewing and once it is working, change it to perspective viewing. The scene has green grass on the ground. There is a yellow sun in the sky. Draw **three** 3-D houses surrounded by grass. Two of the houses are scaled versions of the original whose vertices are the only ones specified in the program. These vertices of the original house being drawn must be around the origin (0,0,0). The look and the relative positions of the houses are up to you.

Create parameterised functions in your program so that your display function (the display callback) calls other routines, which do the drawing.

When specifying the vertices for the polygons you are going to use, code them such that the glVertex calls for each vertex is listed anti-clockwise. For example, when you have 3 vertices: p1, p2 and p3 and when going from p1 to p2 to p3 is anticlockwise then code:

```
glVertex??(p1);
glVertex??(p2);
glVertex??(p3);
```

If it is not anticlockwise, then code an anticlockwise arrangement of p1, p2 and p3. The order is important for later use.

If you haven't already started using polygons, use polygons to draw the faces of the house instead of lines.

Exercise 3.4.
Once you have understood exercises **3.2.** and **3.3.** above, extend Exercise **3.3.** so that a user is able to move around, by pressing some keyboard keys, in the world that you have created with 3 houses.

You may want to examine the "Moving the Camera", "Advanced Keyboard" and "Moving the Camera II" tutorials under the Input section at http://www.lighthouse3d.com/opengl/glut/.

Once finished with these, take a look at the Mouse tutorial.

Exercise 3.5.

Stereo images are produced by creating two images with the viewer in two slightly different positions. Consider a viewer who is looking at the origin but whose eyes are separated by n units. What are the appropriate viewing specifications to create the two images?