Christopher Trimmer

CS-330-T4219 Computation Graphics and Visualization

23EW4 – 04/15/2023

**Final Project Reflection**

3D Scene Development Choices

My scene was made up of objects that represent a pyramid, a small box, a coffee cup, and a plane. The plane is large, flat, rectangular object that represents the surface of the scene. In this scenario, it is a concrete area. The model for my pyramid was based on a rubik’s cube-style pyramid that had varying colors. However, for my scene, I wanted to create a mystical style pyramid. So, I used a brick texture that has specular highlights in varying crevices between bricks. As lighting passes over the pyramid, the specular highlights have a glow, which accentuates the mysticism of the pyramid.

The small box was intended to look like an unopened crate. The texture is made of wood, so I used more diffuse lighting instead of specular lighting on this object. The coffee cup is made up of a sphere and a small-pinch style handle. Most coffee cups are shaped like a sphere, so it made sense to use that shape. The cup also has specular lighting since it is a shiny object.

As we learned in the course, we use triangles to create rectangular objects in OpenGL. So, the plane and the small box are both made by using two triangles for each side. The plane only required two sides, but the small box required 12 since it is actually a cube. The pyramid was the easiest object to create because I only needed the three sides, and a rectangle (made from two triangles) as the base of the pyramid. The coffee cup was the most complex object, and it was made up of a sphere and another small box.

In OpenGL, a sphere can be created in several ways. The approach I followed is from Song Ho Ahn (2023) where adjacent vertices are triangulated to form polygons. The alternative approach would have been to use single triangle strips in a loop to render the whole sphere, and we would still need to deal with the top and bottom. One problem with this approach would have been dealing with normal and textures if there were shared vertices. By following Song Ho Ahn’s approach, I was able to render a very nice-looking sphere to represent the coffee mug (OpenGL Sphere).

3D Scene Navigation

There are two input devices in my scene: the keyboard and the mouse. These input devices enable the user to navigate the 3D scene by controlling the camera. The mouse cursor is used to control the orientation of the camera. The user can look up, down, left and right as if controlling using the mouse cursor (even though the cursor is not in the scene). Most mice have a middle scroll button, and this is programmed to adjust the speed that the camera travels around the scene. The keyboard is used to control the forward, backward, left, right, upward, and downward motion. The following keys are programmed:

ESC Exit the program

P switch view from Orthographic projection and Perspective mode

W move forward

A move left

S move backward

D move right

Q move up

E move down

Code customization and modularization

I feel that I have developed a modularized and well-organized application for this project. First, I made sure to separate the various file types that are used in the project. This includes separating header files, implementation files, and assets into different folders. The assets folder is sub-divided into folders for textures and shaders. This helped in keeping the structure of the project well organized and logical.

From a programming perspective, I created classes for the main objects in the program. This included classes to represent the objects that control the application, such as the camera, lighting, and the application itself. I also created classes that represent the shapes and objects in the application, such as the textures, shaders, meshes, and the project objects. I use an abstract class to represent project objects at a high-level, which I then derive child classes to represent the pyramid, the small box, the sphere, and the plane.

References

DeVries, J. (2016). LearnOpenGL - Introduction. Learnopengl.com.

https://learnopengl.com/Introduction

‌OpenGL Sphere. (n.d.). Www.songho.ca. http://www.songho.ca/opengl/gl\_sphere.html