Paul Kim

CS 330

Final Project Reflection Paper

My 3D scene consists of a computer monitor, a cup, book and a bottle. I decided to go with these objects because I believed them to be great choices since they have a variety of shapes needed to recreate them in OpenGL. After many attempts and researching on how to create these required shapes, specifically the cylinder, I finally was able to program for the required functionality.

Proposed Scene



Recreated 3D scene



To recreate my 3D scene, I had to craft various geometric shapes including a circular plane, a cube, a torus, and three types of cylinders: a standard cylinder, an open cylinder, and a conical frustum. Designing the cube was straightforward; I employed vertices and indices to construct it using triangles with ease. However, creating the cylinder and the conical frustum proved more challenging. The process involved employing trigonometric functions to compute the vertices and indices for each face of the cylinder. Through iterative steps, these functions determined the positions of vertices along the sides of the cylinder, utilizing cosine and sine functions to calculate the x and z coordinates based on the angle around the cylinder. Ensuring proper lighting effects and texture mapping required careful calculation of normals and texture coordinates for each vertex. The generation of indices for the top, bottom, and lateral faces of the cylinder necessitated consideration of the cylinder's perimeter wrapping, demanding extensive research and trial and error to achieve accuracy.With the cylinder mesh as a foundation, I adapted it to create the conical frustum with slight modifications. Likewise, the open cylinder was derived from the conical frustum by setting the top and bottom radii to the same value. Once I had mastered creating the cylinder, producing the other shapes became more manageable. Lastly, for the torus, I employed trigonometric functions to compute positions, normals, and texture coordinates for each vertex. By iterating through major and minor segments and determining angles accordingly, the vertex attributes were established. While crafting these meshes required significant research and calculation, the satisfaction of completing them was undeniable.

Setting up the camera was a relatively straightforward task, thanks to prior modules. I integrated user navigation controls utilizing the WASD keys for movement, and Q and E for ascending and descending. Moreover, the scroll wheel seamlessly adjusted the camera's speed. In addition to leveraging the custom functions from meshes.cpp and meshes.h for creating meshes, I aimed to develop testing functions to assess essential features. For instance, pressing the P button toggles between Orthographic and Perspective displays. Furthermore, I devised a test to switch the lights on and off to evaluate the impact of my lighting setup. Additionally, I incorporated a lamp emitting light within the scene, enabling it to rotate around, providing users with a visual understanding of how the light influences the scene.

In conclusion, this project presented considerable challenges, yet proved immensely gratifying. By assimilating the teachings from the modules and conducting extensive research, I effectively reconstructed my 3D scene.