CS 330 Comp Graphic and Visualization

Final Project

Joseph Caron

12/12/2024

Reflection

The scene I used was a computer desk with a wireless keyboard, a coaster and matching wrist rest, a spiral bound notebook, and a couple of sharpened pencils. I chose this scene because I felt certain aspects of it would offer a challenge to develop into a 3D scene, and yet it felt like a natural scene that could be found in almost any random home with an office. I also wanted to create objects that were complex and made up of multiple different types of basic shape meshes. To meet the required functionality, I was able to load and draw basic shape meshes, and then manipulate them by scaling, rotating and transforming them, then placing them in the proper positions, so that the combined meshes appeared to be one complex object.

In creating the 3D scene, I of course needed functionality for the user to be able to navigate around the scene by manipulating the “camera”. This can be done by using a combination of a keyboard and mouse on the computer. Tilting and panning the camera, that is turning the camera side to side and up and down, is done by moving the mouse. Whichever direction the mouse is moved across the mousepad, the camera mimics this movement. To spatially move the camera, for example moving the camera closer or further from the objects in the scene, or moving the camera in a horizontal or vertical direction, is done by using the keyboard. To move the camera “forward” the user would press the “W” key. “Backward” is done by pressing the “S” key. Horizontal movement is done using the “A” key to move to the left and the “D” key to move to the right. Likewise, vertical motion can be accomplished by using the “Q” key to move upward and the “E” key to move downward. I also made it so that scrolling the mouse wheel in one direction or the other allows the above camera motion to go faster or slower. Finally, I included functionality to switch between a perspective display and orthographic display. Pressing the “P” key will make the screen display in perspective mode, while pressing the “O” key will display in orthographic mode. This set up is designed, as with many computer video games, to make it so the user can easily access all of the camera controls with one hand on the mouse and the other on the keyboard.

I made every attempt to make my program as organized as possible, including cutting out superfluous code. At first, when drawing each basic shape mesh into the scene, I was inputting every aspect of the object in question, such as the scale, rotation and position, even if the information was exactly the same as the last mesh that was drawn. After creating the basic design, I began to cut out any code that did not change a variable in an effort to make the code more organized. I then also changed the order that the meshes were drawn in, so that I could further reduce the redundant code. Every section of code is fully labeled so that anyone who can read C++ code can tell exactly what that part of the code does.