Kayla Kintchen

Professor Enkema

CS-330

8/12/2023

Final Project Reflection

In developing my 3D scene, I made several choices at the very beginning of the semester in regard to how I was going to implement and create different shapes. My plan was to use a cylinder for the cup, a cylinder for the straw, multiple torus for the binding on the notebook, a cube for the notebook, and a cylinder for the speaker. I chose those objects because they are simple, common household objects. While these objects are simple, they meet they are designed with some complexities. I integrated the way the cup tapers towards the lower half, with the straw strategically leaning in the cup. Although I decided against using torus for the binding of the notebook, I decided to rotate and texture a cylinder in it’s place. To then meet my design requirements, I added spheres to my scene in the form of lights.

To navigate about the scene, I implemented camera controls such as mouse movement, keyboard inputs, and the scroll wheel. To use the mouse controls for movement, I record the change in the mouse’s position and convert it to changes in the camera’s orientation. This makes the user feel like they are moving around. The camera movement speed is also directly related to the mouse movement speed allowing the user to customize their navigation experience. I also included keyboard controls such as ‘W’, ‘A’, ‘S’, ‘D’, ‘Q’, and ‘E’. When the user presses on these keys, it navigates the camera around the scene moving the camera forward, backward, left, right, up, and down respectively. Among the keyboard controls is keypress, ‘P’ which switches the display between orthographic and perspective views using the perspective or orthographic projection matrix. The last camera function in this project is the mouse scroll zoom. This allows the user to scroll to zoom in and out of the scene. This is achieved by adjusting the camera’s field of view in relation to the scroll movement.

To make my code more modular, I created different classes for most of the objects and shapes which demonstrates modularity. Isolating the objects in their own classes reduces unintended interactions, creates a clear structure in the project, and enhances the reusability of code. An example in my project would be cylinder class. Because it is separately defined, it reduces the source code file, makes the code easier to read and understand, and can its functions can be reused. Another function in my program that kept the code organized was the loadTexture function. It is responsible for loading image files and creating OpenGL textures from them. This function simplified the code and was able to be reused for every texture. Having a single function for this rather than a block of code for every texture prevents incidents in the code, but also makes the textures easier to manage while enhancing readability.