Frederick Brehm

8/14/2024

CS – 330

Southern New Hampshire University

In choosing the 3D scene that I wanted to replicate I had to make a choice as to the type of objects I wanted to successfully create the scene. Out of the 4 choices that were given I wanted to replicate the desk. I chose the desk over the others as it seemed like a great starting point for someone who has not had a ton of experience with OpenGL. It was not the easiest, but it also was not the hardest. I also wanted a scene where I could gain experience using an abundance of the available shapes. I chose the objects I did in the scene because those objects made the most sense. For instance, when creating the monitor, it was easier to utilize a box shape for the monitor. For the monitor itself I used two boxes and adjusted the scaling of x, y, and z to create the white border and the screen. For the stand I did the same except it was split into 3 different parts, the base, the stand, and the connector. For the books they were boxes that were elongated and rotated. The pencil case was the hardest as I had to use cylinders and boxes to create the finished look. The mouse was a simple sphere, and the keyboard was again an elongated box, with then individual boxes coded onto that to create the keys. Along the way I added textures and shader materials to each object as it was easier to create each piece of the scene if I was able to see how it was going to look visually. The functionality is that the coding for this scene itself is quite simple. Each section is broken down into sceneManager, main code, and viewManager. This allows for easy commenting and coding best practices while maintaining an organized yet functional 3D scene and code.

The virtual aspect of the scene was very easy to create. For basic key movements we utilized WSAD. W was used to move forward, S used to move backwards, A to move to the left, and D to move to the right. Two more keys were added to create a better viewing experience for everyone, Q to move up, and E to move down. Throughout the course we also learned how to access the camera.h file that contained the mouse and navigation controls. Here is where we were able to ensure that the viewers’ capabilities inside of the scene would not be restrained. You can look in any direction 360 degrees, as well as the ability to control the camera speed. When utilizing both the viewManager and camera.h, this allows for the maximum viewing experience possible.

In this project there were several functions used to create a simple yet exciting viewing experience. In viewManager, each function was organized by similarities. If I was working on key movement then all the key movement functions were together. If I was working on camera or navigation movement with the mouse, these functions were together. This helped create code that was organized and easily commented. This approach was also used for the sceneManager. SceneManager consisted of texture, material, shader, light, scene prep, and rendering of objects. Each section was organized to contain functions of its own elements. For example, in the shader portion of the code, SetShaderTexture, SetShaderMaterial, and DefineObjectMaterials were grouped together as they are part of the shading aspect of the project. This approach would remain the same for each component in the code. This allowed for optimum functionality, organization, and best practices.