Dante Lee

16 June 2023

CS-330-J7562

3D Scene Project Reflection

**Justify development choices for your 3D scene**

Above is the scene that I recreated with OpenGL, containing 4 unique 3D objects including a toilet paper roll, a lock, a tiny skateboard, and an apple pencil. These are a good choice because two of them, the pencil and the roll, are made of two shapes, while the skateboard and the lock are made of 4 or more. This will give me a couple easier ones, and a couple more challenging ones to create.

The toilet paper roll is made of a cylinder with another cylinder through the middle, which creates the cardboard roll as well as the paper wrapped around it, and the extra bit of paper that is unrolled from it is a simple rectangular plane. The pencil is simply a long and thin cylinder, with a pyramid at the end. The skateboard is composed of one long, thin box for the deck, with two smaller ones connected to it, angled up a bit to fit the edges of the board, and two small cylinders attached to each side for the wheels. The lock is made of one cylinder for the body, a smaller cylinder for the knob, and a torus for the shackle, which will be partially inside of the body.

**Explain how a user can navigate your 3D scene**

The user can navigate the 3D scene by using the WASD keys to change the camera position in the X and Z directions, and the QE keys are used to change the camera position in the Y direction. WASD handles forwards, backwards, left, and right, and QE handles up and down. The mouse can be used to look around in any direction, while the scroll wheel adjusts the speed at which the movement keys apply change in position when pressed, by changing the coefficient of speed in the position calculation. You can also use the P key to switch between perspective and orthogonal views.

These controls allow the user to flexibly navigate the 3D scene, and look around at any of the objects from any angle to examine each complex object made of fundamental shapes. The user manipulates the keyboard and mouse simultaneously, which will feel natural to anyone who has played a first person video game before, and immerses them in the scene.

**Explain the custom functions in your program that you are using to make your code more modular and organized**

There are many custom functions in my project, which serve to separate each part of the program’s execution into their own place. Some of the functions include the mouse and scroll callback, which execute during each frame of the program, handling user input. There is also a callback for switching between perspective and orthogonal views, which is triggered in the same way. The main function includes all of the setup and configurations for OpenGL including loading the textures, vertex objects and additional settings, and then once it is established, the render loop begins and loops by rendering the scene and objects frame by frame until the program is exited.

This organization keeps the program modular and organized, which is especially helpful when adding more objects to the scene, because you can simply go to the section where you have declared every piece of code that is similar, and add more blocks to flesh out the program. These are all commented nicely and make such a complex program much easier to navigate and visualize from a programming perspective.

**3D Scene**

