­Design Decisions

Jonathon Gaspers

Southern New Hampshire University

Design Decisions

The objects I selected for this project were a random selection of things from my desk. I chose them based on the required functionality for the program; I needed four objects, one that could be made of two or more primitive shapes. The objects I selected were a remote, a vape mod, a ring, and a tiny container of lotion. I chose these because they could easily be broken down into primitive shapes; the remote is a box with a cylinder as an RF transmitter, the vape mod can be represented with a box, the ring as a thin ring, and finally, the lotion container can be created using two cylinders. Initially, I was going to have the remote have the cylinder poking out of the top side to represent a singular button, but I felt it made more sense to use a texture to show the buttons and have the cylinder be the RF transmitter because I could make use of the circular texture the top side of the cylinder allows to represent the “fisheye” shape the transmitter would have. I also chose to split the lotion container into two cylinders stacked on top of each other because after looking closer at the lotion container, the red and white parts were slightly different sizes, and coding a cylinder that had those different sizes would have taken longer than the provided time for the project.

To add camera controls to the program to allow the user to “fly” around the scene to look at the objects from various angles, I used the camera.h provided in the GitHub tutorials. I used this header file to allow more effortless camera movement and the GLFW functions to capture the mouse movements to allow more straightforward navigation in the 3D scene. Additionally, outside of the required functionality, I added the ability for the users to enable wireframe and textured modes by pressing F1 and F2, respectively. I added this functionality because I felt the users might want to see how the various shapes were drawn and help with seeing the two shapes that made up the two complex shapes in the 3D scene.

I tried to “future-proof” the program by moving the mesh generation to its own header and cpp file. This allowed the code to be more modular and easier to read to enable future coders to understand the code and use it more easily. I also decided to move the code that loads the shaders to be in its own cpp and header files to allow the shaders to be more modular. This allowed me to instantiate multiple meshes and shaders in an easy-to-set-up way. By writing the program this way, I ensured that the code was reusable throughout the course.