Andrei Kourouchin

February 19, 2023

7-1 Submit Your Project

I both created and attempted to create a few types of shapes in my design. Unfortunately, I ran into difficulty with both Cylinders and creating cylinders with different base/top radii and stack count. Due to that, I will attempt to submit a revision of this project with an updated count of objects. When creating my eraser, I chose to create two objects, which were inlaid into each other slightly to give the perception that they were one object. One of which was textured as an eraser head, while the other one was textured as the paper that went around the eraser. I believe this was a good design choice for these two objects. These objects are also built from the same functions, along with the charging block, which is made up of three objects, two of which are the wall prongs and the other the plug body. These objects are textured in metal and in plastic to give the appearance of the plug body and shape. I also chose a pencil and a Burt’s Bees tube to recreate, however I was unsuccessful in fulfilling that goal. I chose these objects as I believed they would both give me a challenge to recreate, and that they were interesting.

A user can navigate my 3D scene using the WASD keys which move the camera forward, left, back and right. They can also use QE to move down or up respectively, which allows for a good amount of nuanced camera angles. The user can also use the mouse to both change the orientation of the camera, as well as scroll the mouse wheel to adjust the speed of the movement of the camera around the scene. The user can also use the keys OP to switch between orthographic projection and perspective displays on the screen. Between these three sets of inputs, the user can pretty much view this scene in any angle and with any type of visual projection they like.

There were a few things I did that helped created modular and organized code, one of which is the inclusion of a resources folder to organize all textures into one place. This allows future developers using this codebase to add their own functions and know exactly where textures will. I also created a function that can create a cube mesh by simply passing the UCreateCube function a GLMesh ID.