CS 300 Final- Reflection

By Alex Frankel



We start with a simple scene- a menagerie of random objects from my cluttered desk. I chose these objects so as to present a challenge creating them but not going overboard with complexity (no teapots in this scene).

Here is how it turned out:

A bottle and a box on a table

Description automatically generated

(pretty good for my standards)

Navigating the scene is simple: the program waits for keyboard and mouse events and transforms the camera position based on the directional input. An example would be when W is pressed, the camera’s Z position is decreased giving the perception of the camera moving ‘forward’ relative to its position and angle. WASD for motion, Q and E for rising and falling.

Rendering the objects in the scene were a piece-meal of setting scaling and position vectors, rotational vectors, color, texture, and materials through the shader and then applying that to one of the basic mesh shapes loaded in. This is akin to drawing the scene shape by shape and then overwriting the shader functions when we want to draw a new one. The RenderScene function contains all of these shapes. Reflecting on the process, it would be interesting and even more efficient to break this function down into the different objects and call separate functions to load them.

Some other cool custom functions that we acquired and tweaked throughout the course are as follows:

DefineObjectMaterials specifies parameters of Phong lighting (ambient, diffuse, and specular lighting) and applies them to a given mesh so that it reacts to the light in a certain way. We were given 6 sample types from the start and can create our own in the same way.

SetupSceneLights allows us to fine-tune the position, color, and other parameters of light sources throughout the scene.

LoadSceneTextures and SetShaderTexture allowed us to specify image files in the proejct’s directory and then apply them to different objects in the scene.