Eric Bryant

CS-330 Comp Graphic and Visualization

Southern New Hampshire University

4/16/2023

**Design Decisions**

**Justify Development choices for your 3D scene:**

The 3D scene that I decided to create consisted of random objects that I found around my room. These objects are a water bottle, a glasses case, a box for a ring, and a rubber stopper that you put on the corner of a desk. All these objects were then placed on the desk. For the water bottle I decided this object would be my complex object. The base of the water bottle was made from a cylinder and once the object started to taper towards the top, I placed a tapered cylinder on top of the bottle base. The water bottle still needed a cap, so I created a much smaller compressed cylinder to place on top of the tapered cylinder. The box for the ring was made from a cube and the rubber stopper was made from a prism. Lastly the glasses case was also a cylinder that needed to be scaled correctly to be compressed to take the shape of the case that is displayed in my picture for the 3D scene.

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

A user can navigate my 3D scene by using multiple inputs on the keyboard and mouse. OpenGL has a camera.h file that allows the creator to make basic movements. Parts of the file were edited and improved to add additional functionality. For example, the camera.h file had the WASD keys move forwards, backwards and left and right, but what about up and down? First, you have to add the keys to the camera.h file that you want to apply the up and down movements to. For me, I used the Q and E keys for up and down movements. After updating the camera.h file I could then add the following code to my UProcessInputs function in my source file to properly apply the up movement using the Q key.

if (glfwGetKey(window, GLFW\_KEY\_Q) == GLFW\_PRESS)

gCamera.ProcessKeyboard(UP, gDeltaTime);

Additional functionality includes using the scroll wheel on the user’s mouse to zoom in and out, and also when moving the mouse, it will change the orientation of the camera for more precise movements and viewing capabilities.

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

One function I have is to compute the lighting model using the Phong reflection model. This function will calculate ambient lighting color, diffuse lighting, and specular lighting. This function will also mix the Phong result with a texture or object color.

The URender function is where I can control the objects that I am creating such as the water bottle, plane, glasses case, key case, and rubber stopper. From here I can control the projection and view matrices and specify the attributes of the object to be rendered. For example, If I am creating a cylinder, I can control the rotation of the about, translate the object to a specific place in my 3D scene, and scale the object. Another thing I can do is define if the object has texture or not. Finally, I can initialize the drawing command to the GPU.