Project Reflection

Steven Anderson

When deciding what I wanted to use in my scene I looked for objects that had distinct shapes. I chose the two containers since they could provide a good base for the image being cubes. Then I wanted something round to try working with spheres. Which lead me to choosing the orange. I also thought about making some things challenging for myself, which is why I ended up adding the bottle and the hat. The bottle since it was a label on glass with a topper allowing me to work with cylinders along with layered textures. Finally, the hat, it was my challenge object since it sat at an angle on top of the container and would require 3 objects to create.

The containers I ended up making with two cubes each. One for the clear bottom portion and one for the top lid of each container. I applied the layered textures to make a translucent blue glass texture and used an online plastic texture for the lids. When making them I made the lids 0.001 bigger than the containers so the lid textures wouldn’t clash with the container textures. Also, when making the rectangle container I was able to just up the y coordinate in scale to make the container taller.

While creating the bottle I ending up learning how to apply the layered textures since I needed two for the bottle. I also while creating the bottle realized that I needed a separate texture for the bottle top since the wrap around layered texture I used for the neck of the bottle wouldn’t cover the entire top. The orange was not much of a challenge after making the containers and bottle. I just had to make sure it was positioned right and had an orange peel texture applied. Though lighting the orange was more of a challenge, it required changing the specular lighting so that the rear light and front lights didn’t clash with each other around the orange.

The hat I ended up creating using a sphere for the top, a cylinder for the main center section, and another flattened cylinder for the brim. I also had to layer textures to get the leather strip around the central part. The hardest part of the hat was positioning it right since I had to do micro adjustments to the sizes of the parts and the angles to get them to line up right. With the sphere requiring changes in the 2nd decimal point for both scale and translation.

The navigation in my scene is controlled using the A, D, W, and S keys for move left right forward and back. Then the Q and E keys are used to move up or down. The mouse is used to control the camera angle and by processing its movements. The O key puts the scene into orthoscopic view while the P key returns to perspective view. This was done by altering the camera.h header file to include an OrthoView function that locks the camera in a pre-set place and sets the view to orthoscopic. I also put the A, S, W, D, Q, and E key commands inside an if statement that checks to see if the scene is locked in perspective view. Speeding up around the scene can be done with the mouse scroll wheel which increases or decreased the speed.

In my program I did create a lot of modular parts that could be pulled and used again. I both made and destroy the shader in their own called functions. Along with doing the same for textures. My process input can be used in other OpenGL programs that use similar or the same camera.h header file code. I separated every object being added to the scene with dividing comment lines to better read when each item is being created. I also organized the naming throughout the file to bundle similar object declarations together like my texture ids. The texture ids themselves are all grouped in order in my main function followed by their bindings being kept in the same order. My render function also is divided in a way to try and group as many similar objects together and follows an organized layout for when each item is assigned.

In my creation of new objects, I start each with their bind vertex array line followed by their scale, rotation, translation and model assignment lines. Then this is followed by declaring if it will or will not have a layered extra texture followed by their texture assignments. Then their light assignments declaring the specular light color, location, intensity and size from both light sources for each object. Then depending on if the object has layered textures or not I either move straight to drawing the whole object or break it down and reassign textures as need by while drawing each part of the object. Finally, I end each object by clearing the bind vertex array.

Another organizational part I put into my program was when I found out that the OpenGL we were using holds up to 16 textures, so just in case I ended up needed to many textures I labeled as I loaded each texture what number they were. I also made sure to comment as much as I could so if I scrolled to a random part of my program I could tell where I was from the comments on screen. The only things I wish I could have figured out more was blending and material properties.