Nathan Chuluda

[CS-330-T2653 Comp Graphic and Visualization 21EW2](https://learn.snhu.edu/d2l/home/897781)

Project One Reflection

December 2, 2021

For this project, we started by submitting a photograph of real life objects that we would be tasked with modeling in the course. When selecting these items, it was important to choose figures that we could model using at least four different shapes. The shapes we were encouraged to choose from included a plane, cube, pyramid, cylinder, cone, sphere and torus. At least one item we selected needed to be complex and require several shapes placed in relation to each other to be modeled accurately. I chose the softball for a sphere, the masking tape for a cylinder, the funnel for a cone, and the can of tomato paste as an additional cylinder. As I learned in the course and was in the process of modeling my objects, I found a small metal flask that I thought would be interesting to model. I was able to use cylinders to create the rounded edges of the flask. These cylinders were joined by a rectangle, and then several additional cylinders were placed on top of these to form the cap and spout of the flask. This was the most interesting item I ended up including in my scene, and I enjoyed problem solving how to arrange the various shapes it is composed of.

To navigate the scene, keyboard and mouse controls have been implemented. Almost immediately, a user will notice that moving their mouse will change the direction that the camera is facing. This allows the user to glance left, right, up or down without changing the position they are currently viewing from. Next, the keyboard controls allow the user to move within the scene. W, A, S and D are used for left, right, forward and backwards movement, similar to how the arrow keys are laid out on the keyboard. The Q and E keys allow for movement up and down in the world. By scrolling on the mouse, the user can change the speed at which they move within the world. The movement controls are highly customizable within the code, and could readily be adapted for different input devices. For instance, a joystick could be implemented instead of the keyboard for movement within the scene.

To keep the code base clean and organized, several functions are used to modularize the code. An immediate example that comes to mind is the code for modeling cylinders and spheres. These are separated in header files which are imported to the main source code file. Each time a new cylinder is needed in the scene, it is possible to call this code and instantiate a new object. Modeling a cone is similar to modeling a cylinder, except the bottom radius is not the same as the top radius. I wasn’t able to alter the cylinder header file to accept this additional parameter, so I wrote a function to imitate a cone by stacking a series of consecutively smaller cylinders. This drawFunnel() function is not the best way to model a cone, but for this course, it was a creative attempt on my end, and serves its purpose well. By increasing the number of stacked cylinders, it can give the illusion of a smooth edge of a cylinder. The problem with this approach is that each layer requires an additional cylinder, which is not efficient. As the number of cylinders increases, the performance of rendering the world decreases. There is also a problem with this approach to modeling a cone, as the normal values are not correct when it comes time to add realistic lighting. My scene uses a metal texture to imitate the way light would reflect off a curved metal surface. Overall this project fluctuated between an enjoyable creative process and periods of frustration as I learned my way around the OpenGL code.