Project Reflection

When I first started this course, I knew that the one thing that I wanted was to get the most out of it as I could. That is why I selected to develop the objects that I had selected, which was a jar that I made from two cylinders, box that I made with a cube, pyramid that is made with six triangles, and a sucker that is made with a cylinder and a sphere, I also had a plane to set all my objects on to set the scene. I knew by selecting those objects I would be able to use almost all the primitive shapes. The only primitive shape that I did not use was the torus.

void processInput(GLFWwindow\* window)

{

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

glfwSetWindowShouldClose(window, true);

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

cameraPos += cameraSpeed \* cameraFront;

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

cameraPos -= cameraSpeed \* cameraFront;

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

cameraPos -= glm::normalize(glm::cross(cameraFront, cameraUp)) \* cameraSpeed;

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

cameraPos += glm::normalize(glm::cross(cameraFront, cameraUp)) \* cameraSpeed;

if (glfwGetKey(window, GLFW\_KEY\_Q) == GLFW\_PRESS)// Press Q to move the camera up the y-axis

cameraPos += cameraSpeed \* cameraUp;

if (glfwGetKey(window, GLFW\_KEY\_E) == GLFW\_PRESS)// Press E to move the camera down the y-axis

cameraPos -= cameraSpeed \* cameraUp;

if (glfwGetKey(window, GLFW\_KEY\_R) == GLFW\_PRESS) // Press R to reset the camera speed

cameraSpeed = 0.07;

if (glfwGetKey(window, GLFW\_KEY\_P) == GLFW\_PRESS) // Press P to change the view

view2 = !view2;

}

As you can see from the example above, I was able to use the WSAD keys to move around the scene in a forward, back, left and right axis. Once I had that running good in my code, I moved on to using the Q and E keys to be able to move the camera in an Up and Down motion. The user can also use the P key to change the view from 2D to 3D. They can also use the R key to reset the camera speed if they needed to. I added the R key as a camera speed reset when I was having a hard time figuring out how to adjust the speed with the scroll wheel. The user can also use the mouse to look around the scene, when using the mouse, the camera does not move forward, back, left, or right it moves on the yaw and pitch to allow the user to look around the scene.

The custom functions that I have I my code are the cylinder.cpp and shapegenerator.cpp those two functions are how I was able to make the cylinders and sphere in my scene. By doing the cylinder.cpp I was able to reuse it three different times to render three cylinders without having to put a lot of the same code in my main function. The same can be said for the sphere that I made for the top of my sucker. I was going to try and make a sphere in the top of my scene to use as a light source but I still have not figured out how to use light the way that I should be able to.