Juergen Pfau Melh

CS-330

Reflection:  
  
  
  
Navigation:  
  
Navigation for this project was tasked by the professor to implement keyboard control in this project.The controls would be WASD keys as well as zoom functionality and mouse point. I used the resources provided by udemy.com and learnopengl.com  
  
Using the learnopenGL code provided, the WASD Navigation is programmable with the following lines using the GLM library:

**void** processInput(GLFWwindow \*window)

{

...

**const** **float** cameraSpeed = 0.05f; // adjust accordingly

**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;

}.

Generally, the tasks were fairly easy, the <https://learnopengl.com/Getting-started/Camera> guides had more information that I really needed. Some had angular camera settings and more precise movement options, but for the sake of the project, I kept it fairly basic.   
  
Development choices:  
  
My development choices different a lot between the guides I watched and used. First, using code provided by the instructor, i took inspiration from the files. This confused me at times, because the code was jumbled in the github. I took to learning through udemy to set up each function as a class, and decided to have a camera class, object classes, movement classes etc. To make my code more modular and easy to edit and interpret. In terms of development, it was difficult to truly texture my keyboard well, so with the help of 2 toned wood, the keyboard look contrasted enough to set the expectation that the keys were separate from the backplate.   
  
  
Custom Functions:  
Custom functions included the cup, which was combined of several different shaped objects and design choices. The blue skybox was customized blue to hide the fact that I had no “floor”. The reasons for the lack of floor, which would have been a plane was, I wanted to use the table as my plane, and the camera navigating through the table on a plane would have limited the viewing angles for the work i had completed. Ultimately, I think this was the best choice.

Altogether, as a former mechanical engineering student, and using CAD and other 3D modeling software, I have to say, openGL was by far the toughest I’ve had to use!  
  
I see its purpose, it is very accurate and with a skilled user, the product which you can create is incredible, but it is definitely steep in learning curve.