Christopher Ivey

Final Project Reflection

CS-330-T5620 Comp Graphic and Visualization 22EW5

Dr. Loay Alnaji

Southern New Hampshire University

June 18, 2022

A picture containing text

Description automatically generated

I selected a pyramid object for the top of the roof. Roofs are normally in some shape like the one

I used for my image. I used a 3-edged plane that was rotated -95 degrees, for the apartment

walls. Additionally for the complex object of the plants with the plant pots in the room. I used 2

different shapes. I used two cylinders, one for the plant branch and then one for the plant pot.

Then I used a sphere for the plant itself. For the laptop I used two plane shapes that were shrunk

down in translation and scaling then rotated in two separate angles. They are placed in a way to

resemble a laptop.

// render plane backdrop

glActiveTexture(GL\_TEXTURE0);

glBindTexture(GL\_TEXTURE\_2D, texture4);

glBindVertexArray(VAO8);

// calculate the model matrix for each object and pass it to shader before drawing

model = glm::mat4(1.0f); // make sure to initialize matrix to identity matrix first

model = glm::translate(model, glm::vec3(2.0f, 4.1f, 1.0f));// here is

model = glm::rotate(model, glm::radians(-90.0f), glm::vec3(0.0, 1.0, 0.0));

model = glm::scale(model, glm::vec3(20.0f, 10.0f, 20.0f));

ourShader.setMat4("model", model);

glDrawArrays(GL\_TRIANGLES, 0, 18);

Navigating throughout the scene was coded simply with a process input method, that was created in the

code. Also coded in is the mouse controls that allows the user to be able to use the mouse to motion the

camera in various ways. The letter P on the keyboard also toggles in and out the orthographic view.

// process all input: query GLFW whether relevant keys are pressed/released this frame and react accordingly

// ---------------------------------------------------------------------------------------------------------

void processInput(GLFWwindow\* window)

{

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

glfwSetWindowShouldClose(window, true);

float cameraSpeed = 10 \* deltaTime;

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)

cameraPos += cameraSpeed \* cameraUp;

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

cameraPos -= cameraSpeed \* cameraUp;

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

ortho = !ortho;

Each if statement created a check case for the code compiler to test when a user makes an input.

For each letter that was pressed like for example the E. It made the camera go down. When the

letter S is pressed it makes the camera move backwards. I did not use any customized functions

expect inserting various images that I got off the internet in order to add the texture to the image.

This project was amazing, and I loved doing the work.