Homework 4: Texture Mapping

Name: Karla Castello PSID:2138671

April 2023

1 Problem

In this assignment, we use OpenGL and shader to practice texture mapping onto a cube. We will need to write the code for the buffer of the UV coordinates, which are the texture coordinates and write the code to bind the texture. We will also need to finish the code in the fragment shader and the vertex shader in order to output the texture onto the cube.

2 Method

In the main loop of the program, the first thing to do was set up the UV buffer in order to easily access these coordinates. The next thing to do in main is to bind the texture to the active texture unit. In the vector shader, we needed to write code for the vertex position using the projection, view and model matrices. Then, we needed write a code to accept the texture coordinates as a vertex attribute and then forward them to the fragment shader. In the fragment shader, we write a code to ouput the color of the texture at the texture coordinate.

3 Implementation

For the UV buffer code, I first used $glGenBuffers$ and $glBindBuffer$ to generate a memory space and bind the buffer to that space. $glBufferData$ is used to input the UV coordinate data. $glBindVertexArray$ is then used to store the buffer to the vertex array object. Lastly, I used $glVertexAttribPointer$ and $glEnableVertexAttribArray$ to specify the location and data format to the rendering machine. To bind the texture, I used $glActiveTexture$ to activate a texture unit and then $glBindTexture$ is used to bind the texture to a target, in this case it is $GL_TEXTURE_2D$. In the vector shader, I used code from the previous homework to set up the vertex position. Then, I set the output UV as the texture coordinate $vertexUV$ to send to the fragment shader. Lastly in the fragment shader, the color of the texture by using the $texture()$ function that takes the texture object and the texture coordinates as parameters in order to map the texture onto the cube.

4 Results