COSC 4370 Homework 4

Ivan Madrid

Assignment Problem:

Implement texture mapping in OpenGL by transferring the UV data to an OpenGL buffer and completing the code for binding the texture in the rendering loop and the shader to code to draw the texture. The resulting mapping should be a cube with each face having a number.

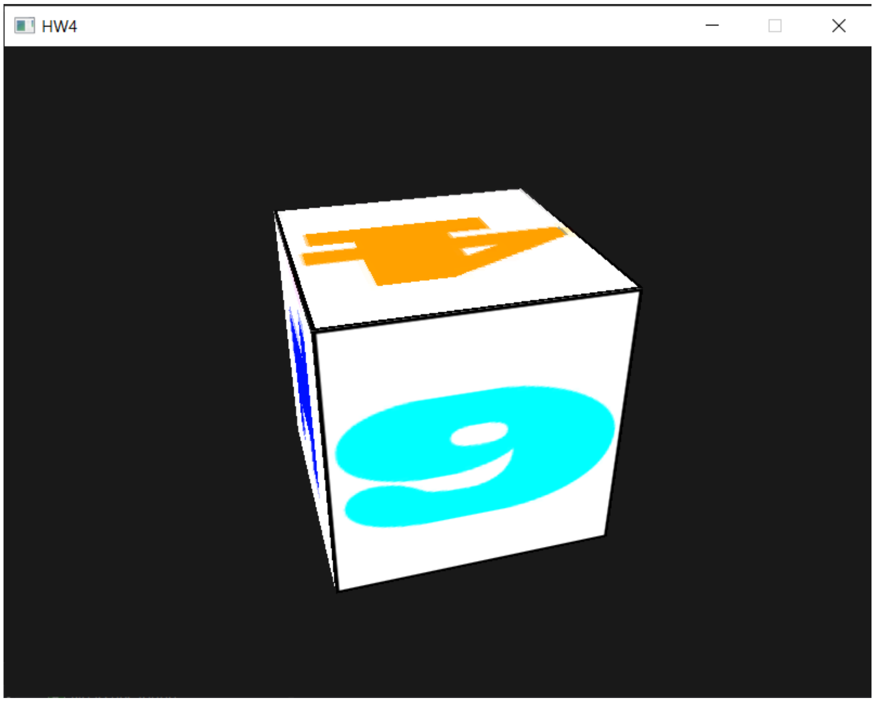
Algorithms and Implementations:

The first step for implementing texture mapping was to create the UV buffer and bind the texture in main.cpp. For the UV buffer, the implementation must generate the buffer, bind the buffer, and import the UV data. The next step was to apply the vertex attribute pointer for the UV data, which is done by specifying the vertex attribute's index, the size, the type, whether the data should be normalized, the stride, and the pointer offset. Then the vertex array must be enabled. After the UV buffer is created, we bind the texture in the rendering loop using the glBindTexture function.

The next step is to complete the fragment and vertex shader to draw the texture. For the fragment shader, all that needs to be done is set the color of the texture function using the texture sampler and the UV coordinates, but the y-coordinate must be inversed for the proper result. The vertex shader outputs the color. For the fragment shader, we set the gl_Position to the projection times the view times the model time the position. This property sets the position of the vertices and gives us the proper output of the cube. Then, the UV is set to the vertex UV, which will be interpolated for each fragment.

Results:

After running the code, the output is a spinning cube where each face is assigned the proper texture with a number on it:



The texture mapping for the cube is correct.