COSC 4370 - Homework 4

Name: Juan Yanez PSID: 2029097

November 2022

1 Problem and Objective

The assignment explores texture mapping in OpenGL. Most of the code is given, however, main.cpp, texture.frag, and texture.vs need completion for the texture to be applied properly on the model. The goal is to have a rotating cube with the given texture applied correctly to each face of the cube.

2 Method

Using OpenGL and some given code, main.cpp, texture.vs, and texture.frag need to be modified. The texture will be mapped by placing the texture coordinates on top of the cube, which will be in the uv array. The biggest thing of importance to have the texture orientation be correct will be inverting the y coordinate points from the UV array.

3 Implementation

With the use of OpenGL's functions, the texture mapping can be done by completing the UV buffer setup, creating the projection matrix inside the game loop, and binding the texture in main.cpp. Texture.frag is used to retrieve the texture and texture.vs sets up gl_position and UV.

3.1 Main.cpp

The UV buffer holding coordinates for placing the texture on the cube needed setup. First glGenBuffers and glBindBuffer are used to generate a buffer in index 1, UVBO, and bind it for holding vertex attributes. The function glBufferData is used to then fill the buffer with the vertex data, which are the uv coordinates. To define the format of the data in the vertex buffer, glVertexAttribPointer is called with the values 1, 2, GL_FLOAT, GL_FALSE, 0, (void*)0), to declare it as float type, fixed point values. The array of coordinates is then enabled so that a separate value from UVBO is used for each vertex by calling glEnableVertexAttribArray. The projection matrix also needs to be set up, the function perspective is called with a field of view of 45 degrees, 8/6 aspect ratio, near clipping point of .01, and a far clipping point of 100. The last change needed is binding the texture to the actual object. This is done by calling glBindTexture which binds the texture to the target, GL_TEXTURE_2D.

3.2 texture.vs

In texture.vs, UV is set to vertexUV. The y coordinate points are inverted to orient the texture correctly on the cube, the x coordinate points remain the same, this UV variable is then outputted to texture.frag. The variable gl_Position will hold the position of the which is set to projection * view * model * vec4(position, 1).

3.3 texture.frag

The file texture.frag just needs to output the “color” variable which will use the function call texture to sample the textels from myTextureSampler at the UV coordinate. This final step allows for the mapping of the texture image to the cube object.

4 Results

The results require a script to be ran in order to create a 3d environment where the cube can be rendered with its applied texture as the cube rotates.