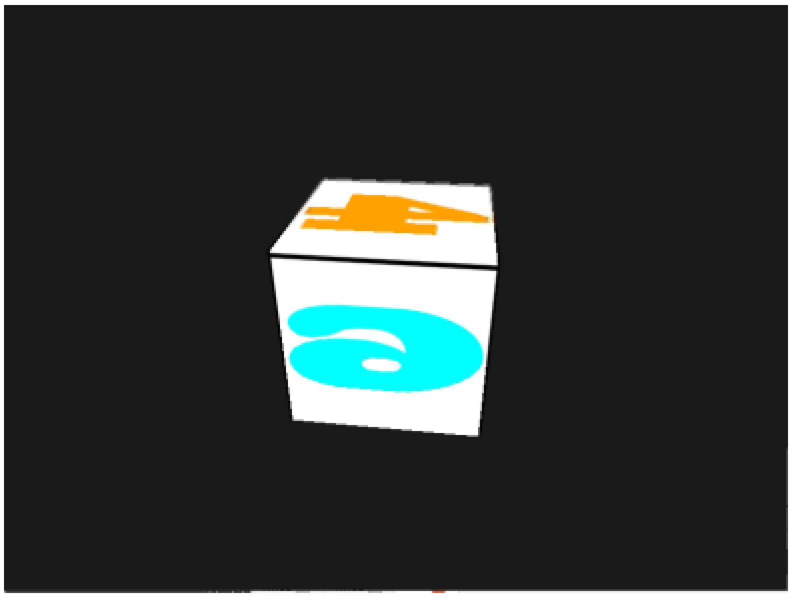
COSC 4370 - Homework 4

Name: Jose Ricardo Sanchez Gonzalez PSID: 1891683

November 10, 2022

1 Problem

The goal of this assignment is to implement texture mapping in OpenGL. Writing the code to transfer the uv data to OpenGL buffer. Also, writing the code to binding the texture in the rendering loop and shader code to raw the texture. By implementing it, the goal is to reproduce a rotating textured cube like the following:



2 Method

First of all, we had to prepare the environment. All we needed was given to us in the files attached with the assignment. We had to fork a replit, and upload all the necessary files. After setting it up, we were ready to begin the assignment.

3.1 Implementation

First of all, in main.cpp, the first step was to set up the UV buffer. It is a step necessary to be able to implement the texture rendering. If done wrong or just not completed, the cube might be incorrect displaying for example, a gray color with no texture. The code used to set up the UV buffer is:

```
glGenBuffers(1, &UVBO);
glBindBuffer(GL_ARRAY_BUFFER, UVBO);
glBufferData(GL_ARRAY_BUFFER, sizeof(uv), uv, GL_STATIC_DRAW);
glVertexAttribPointer(1, 2, GL_FLOAT, GL_FALSE, 0, (GLvoid*)0);
glEnableVertexAttribArray(1);
```

Setting up the projection matrix is the next step to be able to see the cube with the correct shape. The code I used is:

```
projection = glm::perspective(45.0f, 8.0f/ 6.0f, 0.1f, 100.0f);
```

And finally, the last step in main.cpp is to basically bind the texture:

```
glBindTexture(GL_TEXTURE_2D, texture);
glBindVertexArray(UVBO);
glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0);
```

3.2 Implementation

The next file to modify is the camera. The only thing we have to do here is to return the view matrix calculated using Eular Angles and the LookAt matrix. All of this is done inside GetViewMatrix()

```
glm::mat4 GetViewMatrix()
    {
      glm::mat4 view = glm::lookAt(Position, Position + Front, Up);
      return view;
    }
```

3.3 Implementation

Next, inside texture.vs, we first have to set gl_Positio correctly to not to get a black screen.

```
gl_Position = projection * view * model * vec4(position, 1.0f);
```

And after that, we also have to set up the UV correctly, or else, the texture output will be shown wrong.

```
UV = vec2(vertexUV.x, 1 - vertexUV.y);
```

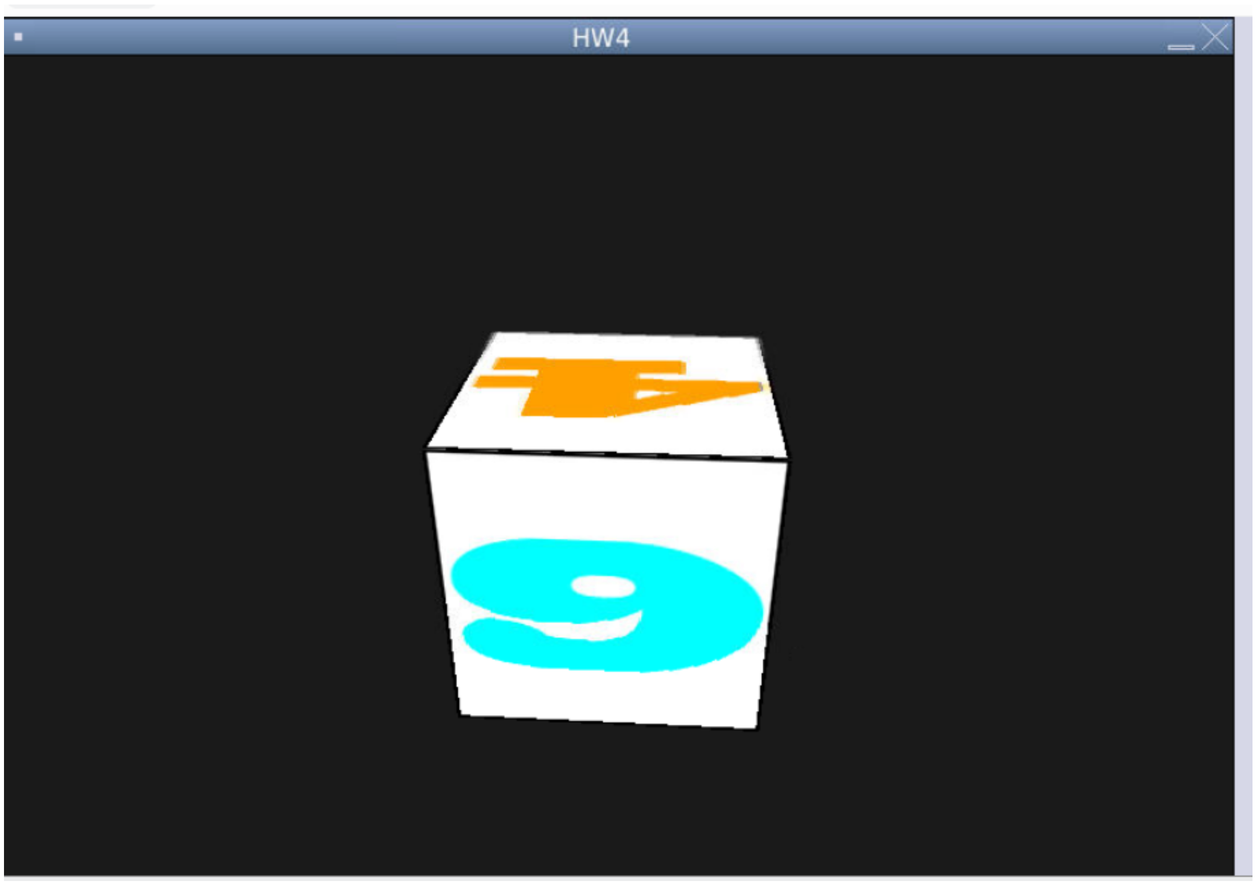
3.4 Implementation

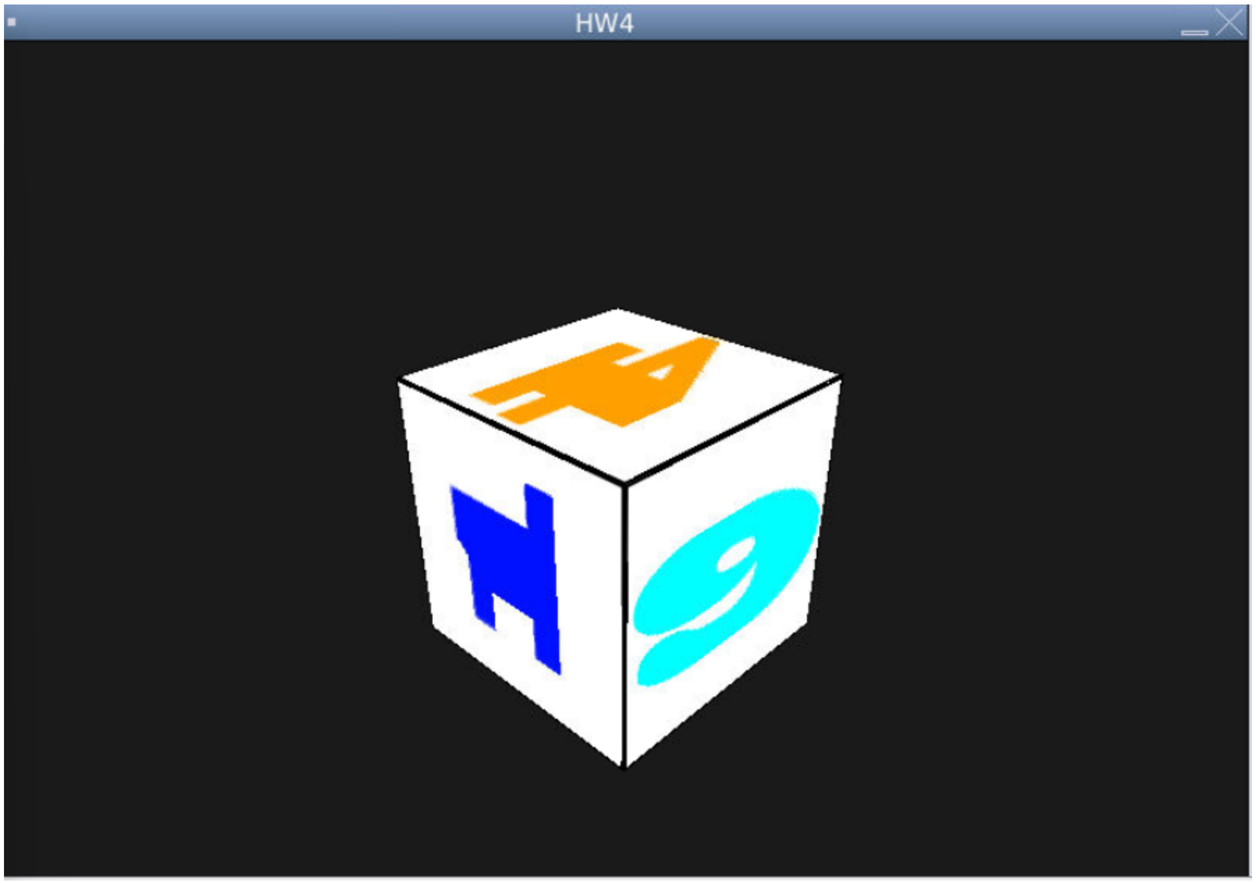
The final step, is to set up the color inside texture.frag. This is achieved by doing the following:

```
color = texture(myTextureSampler, UV);
```

4 Result

The result is a rotating cube with a texture showing a different number in each face. The output is what was intended to get in this assignment.


HW4


HW4