COSC 4370: Interactive Computer Graphics

Instructor: Prof. Zhigang Deng

Name: NHAT NGUYEN

UID: 1652765

HW4 REPORT

Assignment problem:

Practice Binding Texture.

Sub-problems:

- Read, understand, and figure out how to run the given code and get an output.
- Find out which function to use and how they work.
- Figure out how to generate UV Buffer
- Figure out how to get texture.vs and texture.frag to work

First thought:

No new dependencies to install, which is nice.

Same very overwhelming amount of code, but we have some experience with HW3, so it's not too bad. We still have not been explained how the codes work. Instructions given are still not much to work off on.

Process:

main.cpp:

Setting up the UV buffer, follows the process of generating the vertex buffer.

First, we generate the buffer using `glGenBuffers(1, &UVBO);`

Then we bind the buffer to the GL_ARRAY_BUFFER using `glBindBuffer(GL_ARRAY_BUFFER, UVBO);`

We then fill this buffer with data from the uv matrix using

```
glBufferData(GL_ARRAY_BUFFER, sizeof(uv), uv, GL_STATIC_DRAW);
```

Configured it with

```
glVertexAttribPointer(1, 2, GL_FLOAT, GL_FALSE, 2 * sizeof(GLfloat), (GLvoid*)(0 * sizeof(GLfloat)));
```

And lastly, enable the configuration with `glEnableVertexAttribArray(1);`

Note: The `TODO: bind your texture` part doesn't seem to do anything since I can just comment it out and nothing changes. I am sure it does something, but this is high time in the semester so I'm not going to find out.

texture.frag:

I follow the instruction in learnopengl.com, with a bit of tral and error, I got it working.

texture.vs:

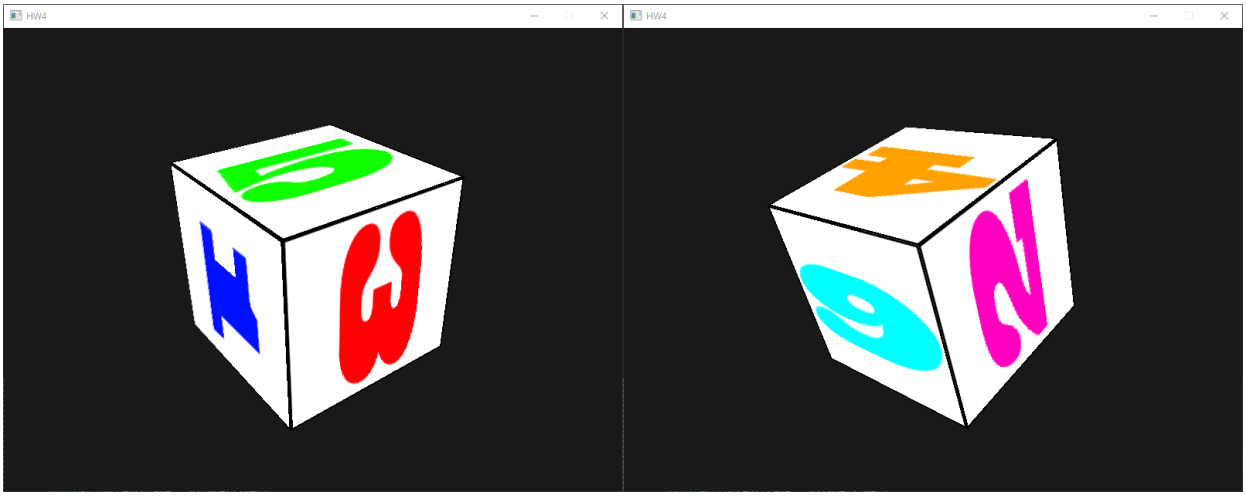
Same as texture.frag, but the result came out inverted, I have to invert it by 1-UV.y to get the result to look right.

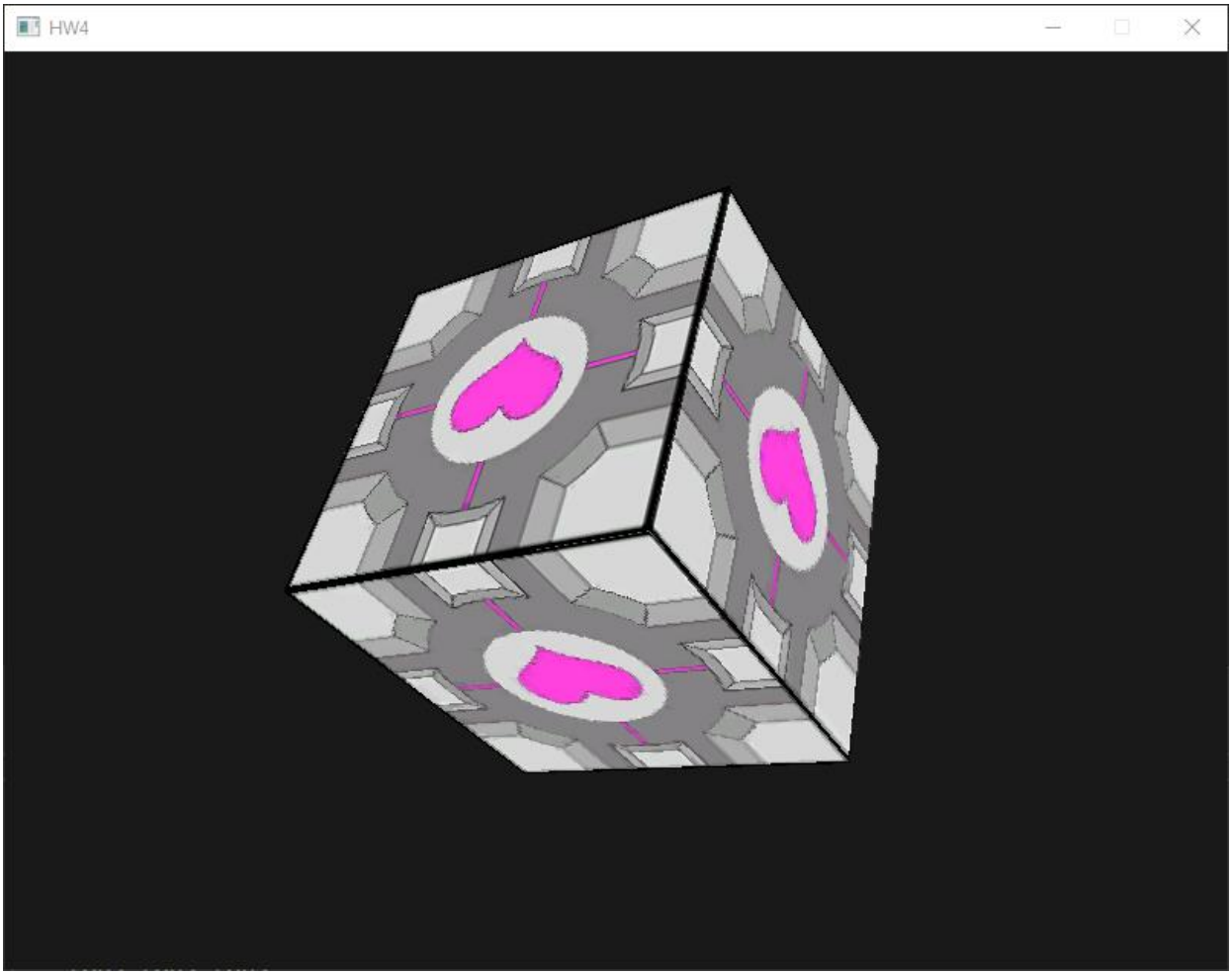
Note: I think it's because of the (1.0f – y)'s in the UV matrix that cause this.

Method using:

A lot of trial and error.

Result:





Conclusion:

I manage to figure out how to get the texture to show up on the cube through some trial and error, but still not really understand how the code or the process works. The lectures and slides help me understand the theories, but not how openGL handles everything.