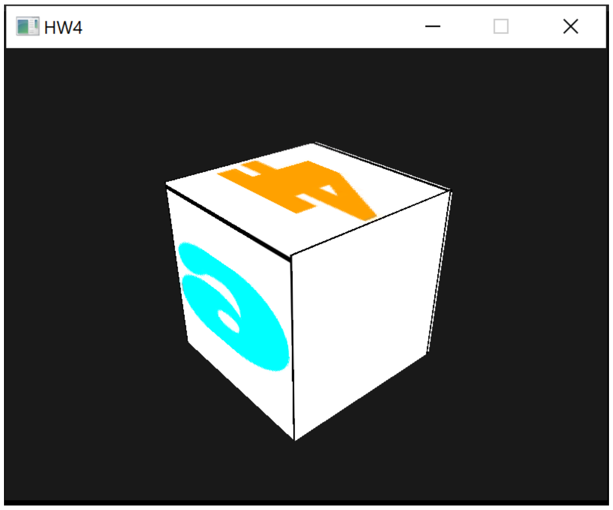
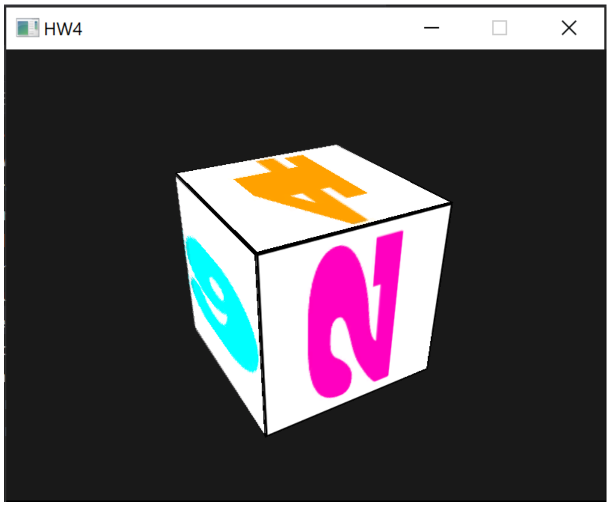
For Homework assignment 4 I was tasked with texturing a rotating cube. Similar to homework 3 I had to code the .vs and .frag files along with the main.cpp. Based on the tutorial on learnopengl.com I first had to start with setting up the UV buffer so that opengl could properly use the coordinates given in the uv array to plot the textures on the cube. To do this I used the section in the code were the VBO buffer was set up. To set up this buffer I had to generate the buffer using glGenBuffers(1, &UVBO). Then after UVBO buffer was generated I had to bind the buffer to using glBindBuffer(GL_ARRAY_BUFFER, UVBO) GL_ARRAY_BUFFER being the target of the bind. We use GL_ARRAY_BUFFER so that we can use vertex attributes. So next I had to fill the buffer with data using glBufferData(GL_ARRAY_BUFFER, sizeof(uv), uv, GL_STATIC_DRAW). After the buffer is filled, now I had to set up the vertex attribute. To do that I used glVertexAttribPointer(1, 2, GL_FLOAT, GL_FALSE, 0, (GLvoid*)0). The first parameter is the index number of the attribute, in this case is 1 because the position attribute was 0. The second parameter is the amount of info in each "section" of data. The 3rd and 4th are standard and the 5th is the stride which is the distance till the next "section". After the attribute was set I had to enable it using glEnableVertexAttribArray(1). Now the code was set up to texture the cube but we still need to set up .vs and .frag.

Setting up .vs and .frag was pretty similar to homework 3. For .vs I used the same gl_Position as homework 3 then made UV = vertexUV. Now that .vs is done I started on .frag which was even simpler than .vs. For .frag all I had to do was make color = texture(myTextureSampler, UV) this was similar to homework 3 as well, just had to using the texture function instead of adjusting the color values. After that I was able to render this textured cube:



This output was the most similar to the sample output in the homework pdf. However, it didn't seem correct based on the texture DDS file and the fact that the white sides of the cube had ugly black borderlines on some of the edges. So, I decided to go for more of a complete textured cube with numbers on all sides of the cube. This was the most complicated part of the assignment because I wasn't sure where the issue was. The Issue ended up being the inputted coordinates in the uv array. Every Y coordinate had 1.0f- this caused all the textures to be "inversed"; which is why the 4 and 6 are reflected in the output shown above. To fix this I simply removed the 1.0f- from all the y coordinates in the uv array. That change my output to:



That was the output that I was looking for. The rotating cube had numbers on all the sides and the black borders were on all edges. Overall, this assignment was fun but there were some confusions I had about the desired output since the original output look closer to the screenshot on the pdf but seemed very wrong. And there was a TODO in the main where it asked for me to bind my textures which never seemed to make a difference to the output. But all in all, it was interesting how much Opengl is capable of doing.