

Texture Mapping

Eric Parsons

2295804

The purpose of this assignment was to understand the fundamentals of texture mapping in OpenGL. We were given hard coded uv data and made to transfer that to the buffer then write code to bind the texture to the render loop and shader code to draw the texture on the cube.

To implement this, first I had to set up the projection matrix which was simple as we did that in the last assignment.

The next thing to do was to bind the texture in the render loop before drawing. I used example code from the OpenGL guide to help implement this. I set the texture ID to 0.

In the Vertex shader I passed the UV coordinated to the vertex shader using the formula $gl_Position = projection * view * model * vec4(position, 1.0)$; (we will come back to this later)

In the fragment shader I set the color to be the texture and used example code from the OpenGL guide.

In the main.cpp, I configured texture wrapping the same way the website explained to do. I used `GL_REPEAT` to wrap the texture properly. I also set up a UV buffer object I could bind and load the UV data to.

It was at this point that I was stuck with a rotating cube that only had two numbers on it. I scoured the internet for answers; why is the cube not matching textures, did I do something wrong, why doesn't my father accept me for who I am? None of these questions were answered. I thought about throwing in the towel. All hope was lost, my grade would suffer, and I would be forced to drop out of college and live in a van down by the river. Then, it hit me. We have a class discord filled with students who are smarter than me. I rolled into the chat, and low and behold the answer was sitting right there. For whatever reason, we needed to invert the UV coordinates with the code `UV = -vertexUV`. Who would have thought? Not me. Anyways, after nearly giving up on life because of a bad texture map, I realized I have finally solved the cube, and now I have working code that does what it is supposed to, and I can once more sleep at night.