Final Project Reflection
Mark Webster

In developing the 3D scene, I chose the Washington Monument area to render. I liked the challenge of using a variety of textures to build the marble areas, grass, and reflection pool areas, as well as planning the layout for the monument and the Lincoln memorial columns (which I partially automated). I knew the plane the render would be on would simulate a grass lawn, and I would need to center the origin in the reflection pool. At each step of the project, I got to see the render evolve. First, I figured out and calculated the vertices and indices for each object. For the indices of the Lincoln Memorial, because there were sixteen columns in total and the pattern that their indices followed was consistent, I wrote a separate "looper" program to generate the indices for the column, saving me a lot of time. Otherwise, all indices were generated manually. The module tutorials helped me implement navigation – including the WASD-keys for basic movments in a flat plane direction, and the QE-keys for movement along the z-axis up and down. I was also able to add the ability to switch between an orthographic projections using the P-key. When the render loads it is presented in orthographic by default, and the user can then press the p key to enter a navigable 3D space.

Once I had the shapes drawn and navigation implemented, I got to spend a lot of time playing with textures. I had to choose textures that were high-resolution enough not to look stretched too much when used on the large surfaces, such as the lawn or the monument faces. Implementing the textures was frustrating at first, but by following the tutorials and YouTube videos, I was able to successfully do so. While there was some stretching, the overall effect was satisfying.

The final step of creating lighting was the hardest of the process. The Module 6 tutorial got me my first light; creating the second was not as easy. The internet is filled with a lot of conflicting OpenGL tutorials, especially ones that espouse deprecated functions or do things that are modern but completely different from how I had my program structured. Building the lighting, like all other steps in this process, was a matter of learning and then adapting principles to fit the program I had written so

far. My first light rotates around the plane, simulating the sun throughout the course of the day. As the "sun" dips below the plane, the scene darkens, and my second light illuminates the scene using a "homebase sunlight" color I found here: https://encycolorpedia.com/fdfbd3. It gives the effect of the scene being lit from the ground up at night by soft spotlights, before the "sun" rises again, the white light and the "homebase sunlight" interacting to form a close approximation of daylight.

I had originally intend for the project to be much more modular. For example, some OpenGL examples I followed on YouTube (including the main one, TheCherno) took great pains to create separate shader files, cpp files for different shapes, *et* cetera. The portability of my program is definitely not there. I was purely focused on the scene render overall than the resusability of code segments. If I could start from scratch, I would follow a much more streamlined and modular approach to the design of the code. Despite this, the main() function is fairly lightweight and functionality is fairly well-contained in the scope of each function. In terms of the artistic quality of the render, the end result is far from incredible—art, even of the digital form, has never been my strength. However, despite the moments of sheer anger and anxiety that were my companions throughout the frustrating journey—my OpenGL PTSD, in fact, should keep my therapist well-paid for eternity—these thousand-odd lines of code taught me a lot about trusting my learning abilities and the power of perseverance.