**Design Decisions**

In my project submission, there are two entities in my scene that I would call objects. The first is the rectangular shape with a custom texture, and the second is the flat plane on which it sits. The rectangular object was created using vertices and indices. I created eight vertices and drew 12 triangles which are defined in the indices variable. The flat plane was made using four vertices and two triangles. Everything is centered at the origin – no translations were applied. I think it would have been easier to create multiple objects if I defined a rectangular 3D shape once and then applied translation vectors to create the same object in more than one place. But this is what I got to work.

I managed to load and apply a texture to the rectangular object. I noticed it was applied to each triangle differently. I tried manipulating the vertices and indices so that the texture would appear as intended but ended up with something crooked. Changing the file format of the texture from .jpg to .png changed the texture from grayscale to color which added some realism.

Users can navigate the scene by pressing the WASD keys as well as using the mouse cursor and scroll wheel. WASD keys move the camera forward, backward, left, and right. This is all done with a function that listens for keyboard input. When a registered key is pressed, the camera Position is reassigned to a new value. Moving the mouse around will orient the camera to face the position of the cursor. When the cursor moves, its previous position and current position are compared. The ProcessMouseMovement function from the camera.h header calculates this and re-orients the camera. Scrolling the mouse wheel “zooms” the camera in and out. This changes the Zoom variable of the camera, which adjust the viewing angle, giving it a “zoom-out” appearance as the angle grows wider and a “zoom-in” appearance as the angle shrinks.

There are no custom functions in my code and that would probably have made it all much more organized and easier to read. I struggled a lot in this class due to having poor foundational understanding of OpenGL. It was pretty easy in the beginning because it didn’t take too much time to change some vertices and complete the homework assignments. But as assignments got more complex, the lack of mastery over foundational topics hurt me. To my credit I will say the class moves pretty quickly and one week might not be enough time to master each set of fundamental topics assigned to us, but if I had to do it again, I would have spent more time reading and re-reading the material at the beginning.