**Matt Knutson**

**CS-330: Computational Graphics & Visualizations**

**SNHU - Instructor Diesch**

**April 17th, 2024**

**Week 7 - Final Project**

For my final project, I ended up adding ten trees in total and a grassy plane for them to sit on. For the trunks of the trees, it was an easy decision to create them using a cylinder. A cylinder can easily be scaled on the y-axis to make the trees taller and shorter. In my original 2D scene, the trees are a bit abstract and appear in the shape of upside-down snow cones. In order to recreate this in 3D, I used a sphere that sits on top of the tree’s trunk, followed by a cone that sits in the middle of the sphere. This gives the appearance that there is one solid shape making up the leaves of the tree. Unfortunately, once I began adding texture, I was unable to wrap the textures around the whole structure in order to give it a fluid design. After completely designing the first tree, I placed the tree trunks in the proper positions throughout the scene. Finally, I added all the tree leaves to the remaining trunks and their known coordinates and gave them each texture.

To move around the 3D scene, the user can use a combination of the keyboard and mouse. The W on the keyboard moves the camera forward, while the S moves it backwards. Likewise, the A moves the camera to the left and D moves it to the right. For added mobility, the Q and E keys were added to move directly up and down. Q moves the camera up and E moves the camera down. Since each of the commands on the keyboard only moves the camera in a straight line, mouse-controlled actions were also enabled. The mouse allows the user to move the camera in a 360-degree rotation while also moving forward, backwards, left, right, up, or down. Throughout the course, I also tried activating the scroll wheel on the mouse to control the speed of the camera movement and tried adding keyboard options to switch between orthographic and perspective modes. I added several different functions to the mouse controls but was not able to successfully get the scroll wheel operational. On the keyboard, I tried adding functionality that would cause the perspective view to switch between true and false when the O and P keys were hit by the user. Unfortunately, I was not successful in this area of the project either.

Within the program, there are several wonderfully intricate functions that are reused throughout its entirety that simplify the 3D building process. For instance, when setting up keyboard movements, the glfwGetKey function is called and used to associate a key with a particular action in the scene. Likewise, the CreateGLTexture function is called each time a new texture needs to be introduced and the m\_pShaderManager function is used on multiple occasions to not only establish a light source, but to also tweak each of that light source’s individual properties. Other functions such as m\_basicMeshes are called only once per item that they need to render. While some functions like SetShaderColor, SetShaderTexture, and SetShaderMaterial are called every time a new shape is drawn in the scene in order to properly define each object.