**OpenGL Project Reflection**

Michael Tricoli

CS-330 Comp Graphic and Visualization

Professor Rodriguez

February 20, 2022

**Justification:**

**The scene I selected was an overhead picture of a paintball field (attached at bottom). The scene consisted of many different shapes in their basic forms. This made it much easier to represent as we were working with developing basic shapes. I was able to create cylinders, trapezoids, cubes, rectangle, and pyramids and a plane within my scene to replicate all the different forms of cover on the field. Most shapes where similar but in different positions, sizes, and angles which allowed me to learn how to use all the different methods for placement that OpenGL has to offer. It also helped to make development easier as I only needed to create five shapes and four textures total to have everything for the scene. This made assigning pointers, VBO’s and VAO’s very easy to do and then just repeat all my shapes inside the render window. Inside the render I was able to create a block of code for each object and assign a texture and different translations, scaling and angles to get the objects where they needed to be. With scaling I had to change around some X, Y, Z to get the objects flat on the surface and not clipping through, but besides that it ran smoothly to bring the scene into a 3D world. While basic coloring would have been fine as paintball cover does not offer much in the ways of texture, I found some nice textured patterns of same color to use to give the scene better aesthetics.**

**Navigation:**

**The navigation used through the scene is the common mouse and keyboard control. You can use the mouse to look up, down, left, and right. These basic movements can also be combining to look up right, down right, up left, and down left. The keyboard allows you to move the actual camera around the scene. The WASD keys are used to move the camera forward, backward, left, and right. The QE keys allow you to move the camera up and down in the world. In addition to the basic movement, you can use the mouse scroll wheel to increase or decrease the speed in which the camera moves through the scene. When you are done maneuvering around the world the Escape key can be used to exit and close the project.**

**Functionality:**

**The biggest help I found while working through this project was the first 3D cylinder project we received. I used this for the base of my project and added all the shapes and textures I needed. The cylinder header and cpp file does all the math work for you and makes creating a complicated shape like cylinder or even sphere simple. All that was needed was assigning the texture, VBO, VAO and attribute pointers then calling the cylinder function anytime you needed one. This allows for any possible variation of a single cylinder to be created and modified with ease as many times as someone would like. All the code that I added for trapezoid, pyramid, plane, and cube was commented in the same areas the cylinder was assigned in the source file. Keeping all the vertex, texture assignments and pointers in same area keeps the code organized and easy to read. I made sure every block of code I added is commented so you can follow and see exactly what each area does what and how much of that is needed to create and place or delete objects. This makes replication or even adding more shapes into the scene easy for anyone to follow. The other important function in the code is the camera header. This allows for all the implementation of camera and keyboard controls to be added into the source code. All you need is to assign the callbacks in the source file, and you can change what the input controls will do when you use them. I was able to take the original zoom function and change it to alter speed when the scroll wheel is used by modifying the header file and just calling back the function in the source file.**

**Original Photo:**

****