Logan Parker

CS330 Project: Mountain-View Reflection

For my 3D scene I was trying to recreate the scene from this picture, which was taken deep in the Rocky Mountain National Park. 

I used a cube for the boulder on the right as it looks very cube shaped. I also used a cube for the boulder on the left, but I also added a pyramid on top. The boulder on the right extends quite a bit farther than how it looks in the picture, so adding the pyramid on top allowed me to extend the bottom while preserving the overall shape. For most of the shapes in the scene I used the same VAO's, modifying them at render time for simplicity. I used the cylinder and sphere shape for the person in the middle, which relied on the cylinder and sphere classes. For some the trees I used the cylinder class and created planes which rotate and climb up the tree to give the illusion of foliage. I would have liked to do this for all the other trees, but to reduce the number of triangles I decided to make the others use pyramids. The pyramid, cube, and plane shapes were all created form VAO's and VBO's in the Source file.

Users are able to control the virtual camera by using typical video game style controls (e.g. wasd, q/e, mouse movements). These controls have long been standard, and it makes since to implement them in this way to help any new users have an intuitive initial interaction. Each frame the program polls the keyboard and mouse drivers via library to see if any inputs have been captured. The changes are calculated through the camera class and the perspective is updated for the next render.

When I first started I created a class called a Scene class. While I believe this class could have been incredibly valuable, I scrapped the idea due to the high level of dependency the class would have with main function. It would be essentially completely coupled and not useful. I decided to add a few more simple functions to help modularize the code, while giving it a more functional programming base. This was simply easier to implement in an OpenGL C++ setting. I think much of the program could be reused, but many things would need to be changed.