My goal for this project was to replicate a 3D version of a 2D image. For my 3D scene, I decided to build a 3D campfire setup that includes a tripod, a cooking pot, rope, stones, firewood, flames, a mountain backdrop, and a forest floor. I chose this scene purely from my boy scout nostalgia. I chose these objects because they are realistic, while still being simple enough to emulate the campfire picture. For example, the tripod legs and firewood are cylinders, the stones are spheres, the flames are cones, and the cooking pot is a sphere that I scaled to look like a bowl. The tripod and pot together count as a more complex object since they use more than one shape to give the full effect. If I were good, I would have figured out how to make the top of the cooking pot concave so that it looked realistic to a bowl. I also would have done more to the flames to look more realistic to a fire rather than a basic cone with a flickering, fire colored effect. The lighting effects added make it look like the fire is illuminating the stones on the inside of the circle.   
  
To navigate the scene, I programmed the camera so the user can explore in every direction. The WASD keys move the camera forward, backward, left, and right. The Q and E keys move the camera up and down. The mouse cursor changes the orientation of the camera so the user can look around in any direction, and the scroll wheel adjusts the movement speed for finer control. I also added a keyboard control to switch between perspective view, 3D, and orthographic view, 2D , which allows the user to view the same scene in different projection modes.

I organized my program by creating separate custom functions for each object in the scene. For example, I made functions such as RenderTripodLegOne(), RenderStone1(), RenderFirewood(), and RenderFlames(). Each function has a single responsibility: to transform and draw one object. This made the code much easier to read and understand, because I can see exactly which part of the scene each function is responsible for. It also makes the code reusable. For instance, if I wanted to add more stones, I could reuse the stone functions with various positions. Organizing my code this way helped me stay focused and keep the project manageable.  
  
Overall, I chose my objects because they are a good balance of simple and realistic, I gave the user full camera navigation controls to explore the scene, and I kept my code organized with reusable functions. This made my 3D scene both functional and easy to understand. By carefully selecting objects that fit together naturally, programming the camera for smooth navigation, and writing organized functions, I was able to create a complete and polished 3D project. The development choices I made were driven by both the project requirements and my desire to make the scene feel realistic. In the end, I am confident that my campfire scene meets the expectations for low-polygon modeling, textures, lighting, navigation, and proper code organization.

References

AmbientCG. (n.d.). Wood 060 [Texture]. Retrieved from https://ambientcg.com/view?id=Wood060

AmbientCG. (n.d.). Metal 029 [Texture]. Retrieved from https://ambientcg.com/view?id=Metal029

docs.GL. (n.d.). OpenGL API Documentation. Retrieved from https://docs.gl/

HDQ Walls. (n.d.). Dusk Mountains 4K wallpaper [Image]. Retrieved from https://hdqwalls.com/dusk-mountains-4k-wallpaper

LearnOpenGL. (n.d.). LearnOpenGL.com. Retrieved from https://learnopengl.com/

Poly Haven. (n.d.). Forrest Ground 01 [Texture]. Retrieved from https://polyhaven.com/a/forrest\_ground\_01

Poly Haven. (n.d.). Rock Boulder Dry [Texture]. Retrieved from https://polyhaven.com/a/rock\_boulder\_dry