Shepherd, James
217257968

The following is a report of my implementation for assignment 1.
This project was quite difficult to get started due to software compatibility issues. Despite that, I chose to continue with this assignment because it was the only strictly VR assignment available (the other assignments were based on AR or AI) and I was particularly excited to try my hand at it.
I was also quite excited by the brief itself. The idea of creating my own environment and implementing my own interactions was an open door to creativity. I felt like I could make it as simple or intricate as I wanted, which really put me at ease as I was relatively unfamiliar with unity and VR specifically.

I decided to implement an outdoor camp sight scene. I felt more familiar with outdoor scenes from the pre-assignments and tutorials that I had seen. I found some simple assets in the asset store and created the scene. I made sure to include water, mountains/big rocks, trees and grassy areas for variety, to make the scene more interesting. I also decided to set the scene at night simply because I prefer an evening camp environment. I used a pre-made particle system for the fire.

I started by adding some simple sound effects. I added a fire/burning sound emitted from the fire, a water sound emitted from the lake and two ambient bird sounds on opposite sides of the scene to provide a natural outdoors feel. I played around with the settings so that the bird sounds would always be audible but the fire and water sounds would only be audible within proximity and that they didn't overlap too much. Ie. The fire sound is more prominent when close to the fire and the water sound is more prominent when close to the lake.

I again made use of the asset store and replaced the player object. The new player object could climb hills without turning the camera unnecessarily, making the viewing experience smoother and more realistic for the client. It also has functionality to walk, run (when holding the SHIFT key), jump (on SPACEBAR) and crouch (when holding the LEFT CTRL key).
When crouching, the speed is reduced.

The player is able to float/swim in the lake.
I tried to find instructions online on how to make objects float in water. Many of the tutorials or descripts I found were more complex than I needed. In the end I decided to duplicate the water surface that I had, dropped it below the original water surface, turned the mesh render off to hide it and added a collider to make it walkable. I then added an animation to make it move up and down slowly. This way it simulates floating when the player walks on it. I duplicated the original water surface instead of using a simple plane so that the player would rise and fall where the waves get higher and lower. I also slowed the player's movement slightly when swimming to simulate the drag from the water.
I was quite proud of how simple this was compared to the instructions I found online.

The rest of the interactions mentioned can only be performed within proximity to the affected objects and have to be invoked by pressing the X key. When one of these actions is available a white ring is displayed on the screen to indicate so and to help the player aim when necessary.

Secondly, the player can chop down trees.
For this I added an axe to the player prefab. The axe contains two animations. One that hides it when it's not being used and one that shows it swinging in front of the player when it is being used.
When the player is close enough to a tree, the white action ring will appear. Aiming this ring at the nearest tree and pressing the X key will initiate the swinging axe animation, remove the tree object and replace it with a tree stump object.

Thirdly, the player can throw small axes at a target near the campsite.
Again, the player has to be near enough to the target, aim the action circle at the target and press the X key. This will instantiate an axe and throw it forward with a given force.
If it hits the target, the axe will become stationary and remain stuck there.
To check whether the player is aiming at something, the program makes use of raycasting. Because of this, if the player is aiming at an axe in the target and not the target itself, the player can't throw.

Lastly, the player can add wood to the fire.
When the player is near enough to the campfire (it doesn't matter where he/she is aiming) the player can throw a piece of wood. This then triggers a particle effect showing the fire momentarily growing bigger and giving off a burst of sparks. The piece of wood is not actually solid, so it simply falls through the scene. After a second or so the wood is then destroyed.