Technical Document

**Proposal:**

The game will start with a beginning menu where the user chooses to start the game or quit. The idea is to make a portal esque game where the user can fire a “portal gun” at walls in order to solve a series of platforming puzzles to reach the next level. This portal gun fires a pair of “Wormholes” that connect the ends of one portal to another for use in the puzzles. These portals will make the user believe that they are in fact connecting two parts of the level, allowing them to use these portals in a variety of ways such as falling through one portal to launch themselves out the other, retaining their speed when either you or an object travels through it. The game will also implement the concept of having objects partially inside the portal, therefore requiring the user to “cut” the object in order to represent this on both sides of the portal. Once the user fires a third portal, the first portal fired will be despawned and the second portal fired will stay and be used as a new entryway. The mechanics of the gun will be similar to that as in a first person shooter, where when the object fired from the portal gun collides with a flat surface of some kind, it will create the portal. The player will also be able to pick up various objects around the scene by using a button on the keyboard. The layout of the game will be several levels consisting of large rooms with various hazards, objects, and platforming segments that the user has to traverse in order to get across the room and enter the exit room that often will require the user to fulfill some kind of condition, such as pressing a button or placing a cube on a pressure plate before opening. If a player falls victim to one of these hazards, the user will “die” and be respawned at the beginning of the level and have to repeat the processes. Some of these hazards will include things like “death” pits and projectiles. There will be no fall damage in the game and the room will be encompassed by an invisible wall in order to prevent the player from leaving the room or going places unintended. At the end of the levels, the user will be given a game complete screen. The player will also have unlimited attempts at the puzzles meaning there will be no real way of creating a “Game Over” besides quitting via the in-game pause menu. The pause menu will also have a restart button to manually restart the level if the user chooses to do so.

**Technical Implementations:**

Standard features:

1. Setup of 3D Virtual Content(Models, sounds, etc.)

2. GUI

3. Startup and In-game menu(Pause, Options button, and quit game button)

4. First person shooting for the portal gun onto various walls throughout the room to create the portals.

5. Hazards(bottomless pits)

6. Implementation of game logic

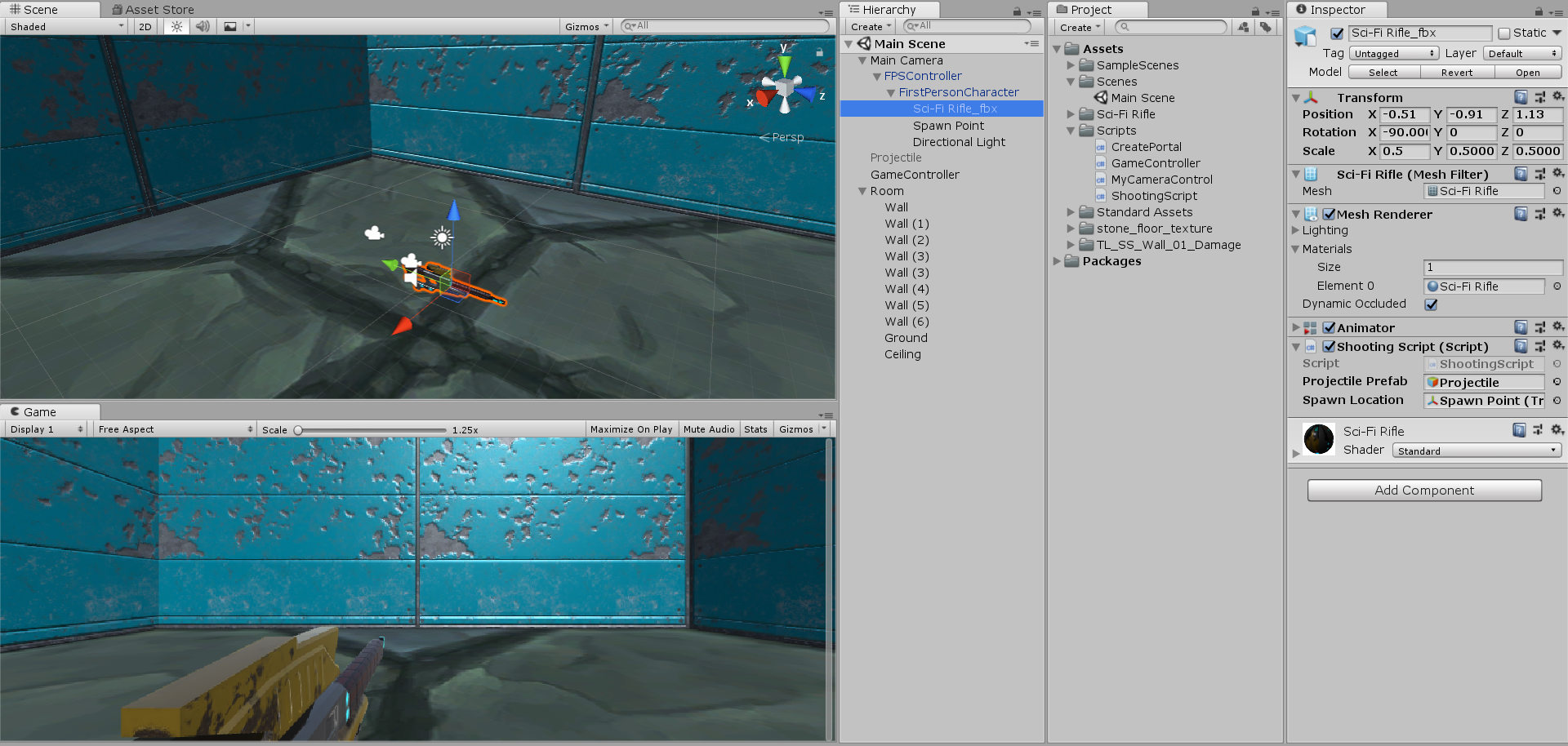
7. 5 Complete levels with significant differences between them

Advanced features:

1. Portal implementation (part 1): Visual effects

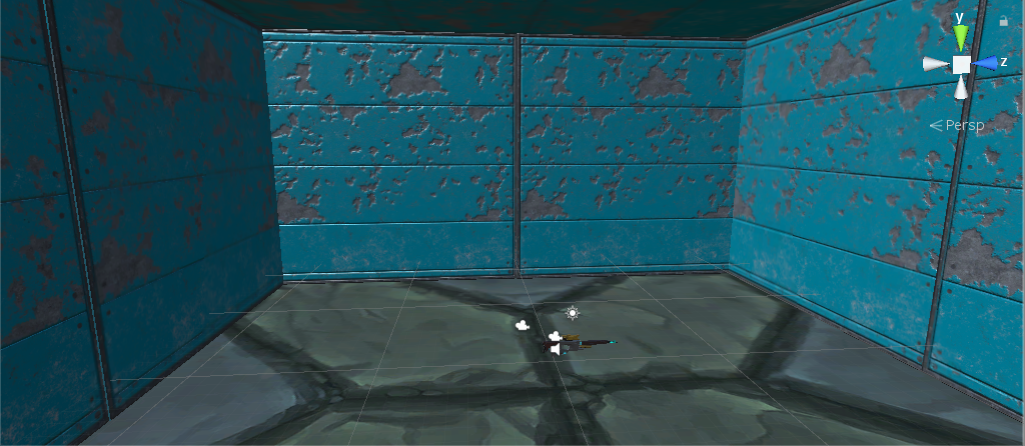
2. Portal implementation (part 2): game mechanics

**Current Progress:**

****

**Figure 1: Current File Layout**

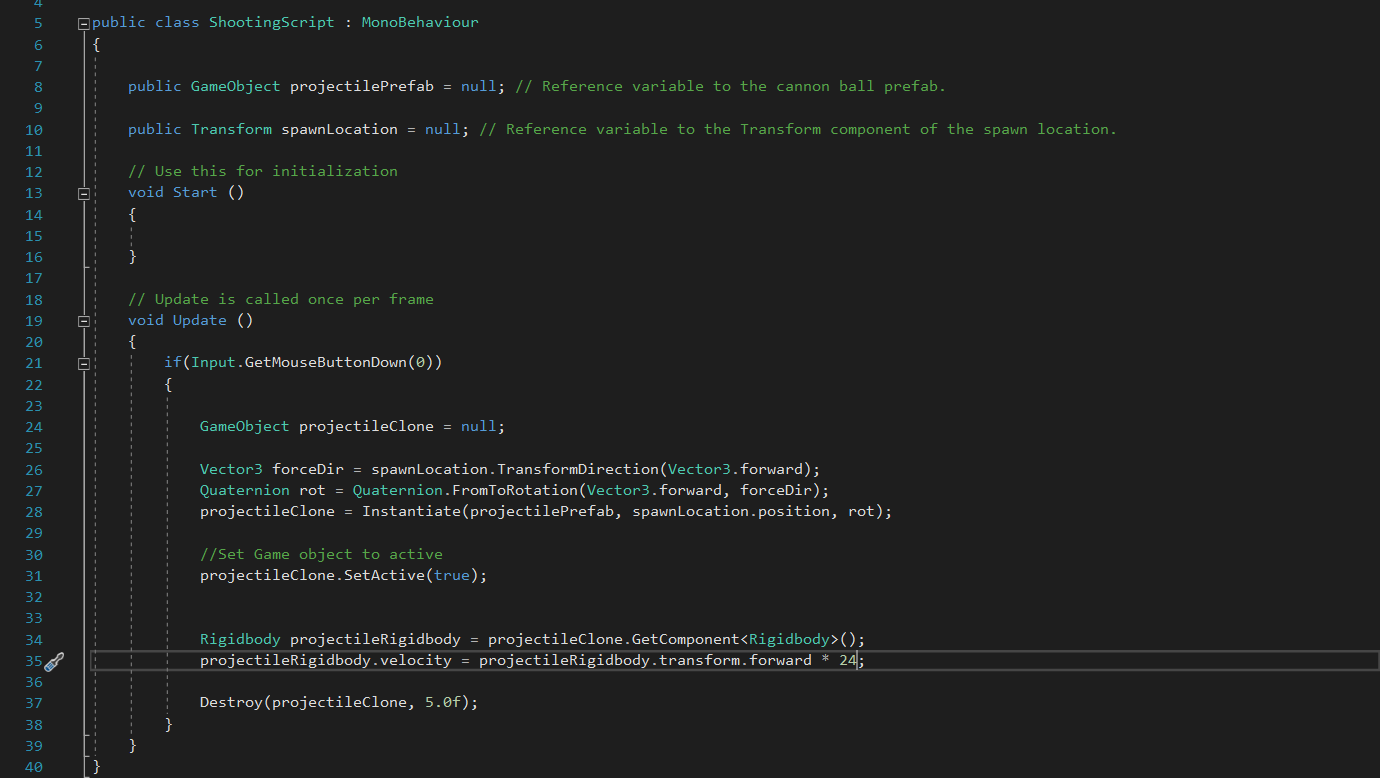
Currently the game is roughly 20% complete. As it stands, the basic fundamentals of first person shooting have been implemented and most of the assets required have been imported. The first thing I did was implement the First Person Controller from the unity standard assets. As you can see, it is nested under the main camera in order to be implemented. The FPS controller also comes with extra things such as audio for walking, and its own rigidbody in order to replicate a player character. The second thing I did was set up the scene.





**Figure 2: Current Scene**

The scene is currently very basic with four walls encompassing the main camera and a weapon underneath the main camera. This is in order to give the user the impression of holding the weapon. In front of the weapon is an empty game object called SpawnPoint that is used later to give the position where the projectiles will be instantiated. The current setup of the walls is only temporary for testing purposes in order to assist in development of the portal mechanics. The next item I worked on was scripting of both the walls and the weapon. The weapon has a script attached to it called “Shooting Script” in order to allow the weapon to fire spheres with force out from the gun barrel.



**Figure 3: Shooting Script**

As you can see, the script activates once the user clicks the left mouse button. Once this is done, a previously instantiated(but deactivated) projectile object is instantiated into the spawn location that is declared beforehand in front of the weapon with the appropriate rotation. The projectile is then set to active in order to finish its spawning. Once this is done, the projectiles rigidbody is obtained and a forward force is applied to it in order to “shoot” the projectile. The projectile is then despawned after 5 seconds. The next script I worked on was the create portal script.



**Figure 4: CreatePortal Script**

Currently, this script is attached to all of the walls in the room and when the projectile fired from the weapon collides with one of the walls, it is destroyed. This script will eventually handle the creation of the portal objects once the projectile collides with the walls. There is one more script that is used called GameController, which will handle most of the games logic, but currently has nothing in it.

**Work to be finished before the next due date:**

1. The portals will spawn when the projectile collides with a wall and will fulfill the basic requirement of teleporting the user from one place to another, potentially with the ability to see the other side of the portal (without the advanced feature)
2. One full room with example hazards should be implemented
3. Start implementing in game pause menu and startup menu
4. Start working on more advanced features of “Cutting” the portal as it enters a portal slowly.