Nanyang Technological University

VR Assignment Report: Technical Innovation

CZ4001 Virtual & Augmented Reality

By: Khiew Jian Bin & Tan Peng Hian

Introduction

This report will explain the thought process, difficulties and the technical innovations of the project as shown during the Live Demo / Presentation. This report will contain animated gifs to better showcase the features. To view the animation, **double click** on the image. We all also be briefly explaining some game design theory.

Assets Used - Imported

Below is the list of assets/packages used in our game

Name	Description	Source
VRTK	Virtual Reality Tool Kit provides support for VR including oculus and HTC Vive. Also provides basic VR interaction scripts	https://assetstore.unity.com/packages/tools/vrtk-virtual-reality-toolkit-vr-toolkit-64131
DissolveShader	Pack of shaders for dissolve effect	https://github.com/kwnetzwelt/unity3d-dissolve-shader
Wolf	Rigged and Textured Wolf model + Animation	https://free3d.com/
Spider	Rigged and Textured Spider model + Animation	https://free3d.com/
45 ACP Pistol	Rigged and Textured Pistol model + Animation	https://free3d.com/
PortalButton – Big button	Textured Portal button model	https://free3d.com/
PushButton	Textured Push button model	https://free3d.com/
Complex Lava/Circuit Ball	Texture pack of Lava and Circuit. Includes normal maps emission maps etc.	http://catlikecoding.com/unity/tutorials/rendering/part-9/
Sci Fi Futuristic Hand Gun	Textured Pistol model	http://gamedevpipeline.com/sci-fi-futuristic-hand-gun/
War FX	Pack of Particle effects for war games	http://jeanmoreno.com/unity/warfx/

Any other gameobjects used that are not part of the above imported assets are created using the base Unity Engine.

Project – Game

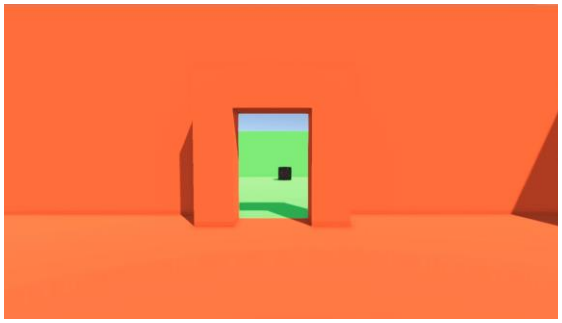
Our VR project is thought as an area/level-based game where each level consists of puzzle/challenges in which the player must complete to unlock an exit door before being able to proceed to the next level. The main objective of the is to navigate and reach to the last level. Each level is more difficult than the previous.

Inspiration

Our game is inspired from Portal and Portal2 which is a 1st person puzzle-platform game where the player must navigate and complete through a series of puzzle rooms with each room is more difficult than the last. The unique feature is the use of "portals" and "portal gun". The portal gun allows the player to shoot two separate "portals" on walls that will allow anything to be teleported from one portal to the other portal. Many elements of our game borrow the same concept used for the puzzles elements and objects in portal.

Initial Phase - Problems

Our first plan was to mimic the game design and features of portal including the "portal" and "portal gun". We used a camera placed on the other side of the portal to render the view as a texture on the current portal. This effect is tested and works well on PC.



Gif 1. Portal - played on PC

When tested on VR, there is a problem. Internally, VR system creates a left and a right game camera from the active camera with offsets to render the image for our left and right eye. The left eye sees a slightly different image than the right. However, the portal camera scripts do not compensate for this, as a result, the left and right eye shows same portal render texture based on the active camera which is wrong and suppose to base on the internal VR left, right camera. We were unable to obtain the internal offset values of the left and right eye cameras for a work around, so we decided to scrap this feature.

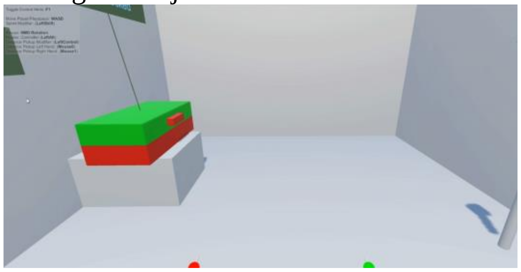
Presentation Time Consideration

Due to the lack of presentation time, we decide to a shorten and compact the puzzle level design into 2 levels to showcase the puzzle mechanics to be used in the game.

Player Movement (VRTK)

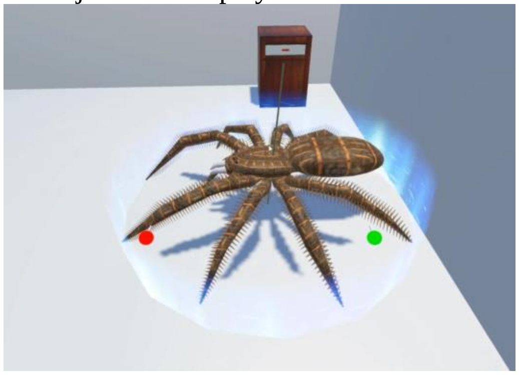
This section explains the different types of player movement mechanics. We plan introduce this to the player as a “powerups” as a way to reward the player for completing certain puzzles and add a way in which the player needs to use new movement mechanic to complete subsequent levels.

1. The player can walk around the room using the VR touchpad(joystick) controller. The player has a rigidbody and a collider to prevent the player from falling though the floor and moving though other game objects.

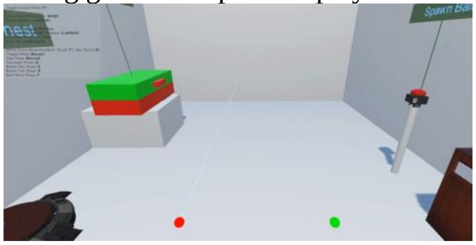
   Gif 2. Player Walking

   If the player collides with a small object while walking, the player will step on top and walk over the object and the player will fall to the nearest ground after walking over the object.

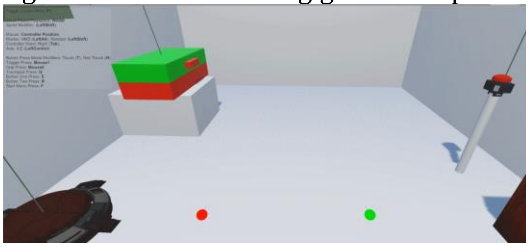
   Gif 3. Player step on spider

2. The player can teleport to the nearest point along a straight path pointed using the left controller. Holding the X button will display the straight path from the direction of the left controller and letting go will teleport the player to the nearest point intercepting the path

   Gif 4. Left Controller Teleport

3. The player can teleport to nearest point following a trajectory path cast by using the right controller. Holding the A button will display a bezier curve trajectory out from the direction of the right controller and letting go will teleport the player to the nearest point intercepting the curve.

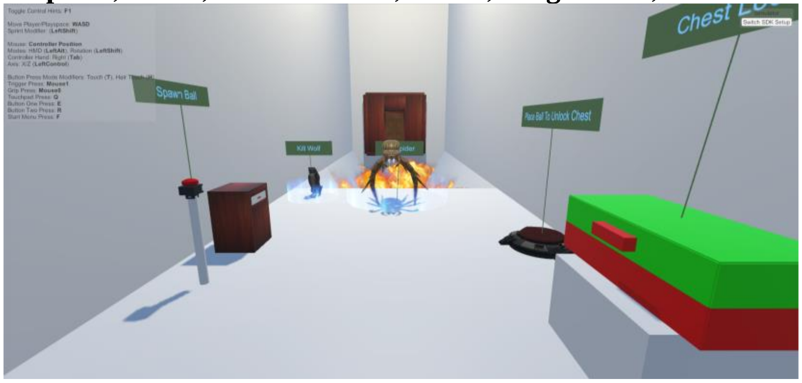


Gif 5. Right Controller Teleport

Level 1

The player wakes up in small room with 4 puzzle elements and 2 animals. The player sees an odd wooden room just ahead and will intuitively believe that that is the exit to move to the next level. There is a large pit filled with fire preventing the player from reaching the wooden room. This presents the first challenge puzzle to the player.

The player can try walking across the bottom of the fire pit but the player will die from the fire and respawn to try again. Due to implementation problems, the dying and respawning mechanic is replaced with invisible collides, blocking the player from moving into the fire.
The player can try teleporting across but will realized that the teleport will work on any floor except wooden material objects.
The player will have no choice but to explore the room to seek a different way.
In the room, there is a **spider**, a **wolf**, a **small button**, a **desk**, a **big button**, **chest and a ladder**.


Toggle Control Hints: F1
Move Player/Playspace: WASD
Sprint Modifier: (LeftShift)
Mouse: Controller Position
Modes: HMD (LeftAlt), Rotation (LeftShift)
Controller Hand: Right (Tab)
Axis: XZ (LeftControl)
Button Press Mode Modifiers: Touch (T), Hair Touch (H)
Trigger Press: Mouse1
Grip Press: Mouse0
Touchpad Press: Q
Button One Press: E
Button Two Press: R
Start Menu Press: F
Spawn Ball
Kill Wolf
pider
Place Ball To Unlock Chest
Chest Lo
Switch SDK Setup

Pic 1. Player Wakes Up

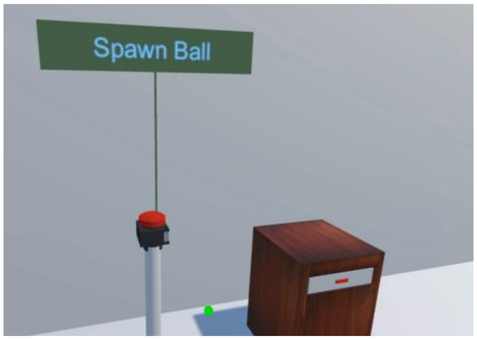
Level 1's puzzle is solved by interacting with and performing the following certain actions using the objects in the following order:

1. Pushing the small button to spawn a grabbable ball (lava / circuit ball)
2. Placing the ball onto the big button to unlock the chest
3. Grabbing and Lifting up the lid of the weapon chest
4. Grabbing Weapons from inside chest and using them
5. Killing the spider that blocks the exit
6. Climbing a ladder

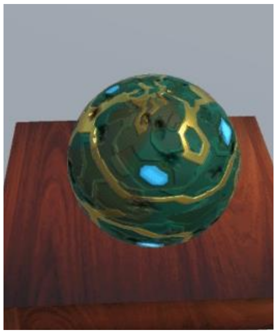
Pushing the small button to spawn a grabbable ball

The first step is to push the small button. The player can use both VR controllers to control the virtual hands to touch the button. When the hands are touching the button, the button will highlight to signify an interaction link and the player can press the VR controller trigger to push the button which will spawn a ball on top of the wooden desk.

The spawn ball type is **randomized** to have a **lava** or **circuitry** material. The ball has a custom script that increments the object uv's offset values over time. This creates an animated shader with the effect of a spinning ball.



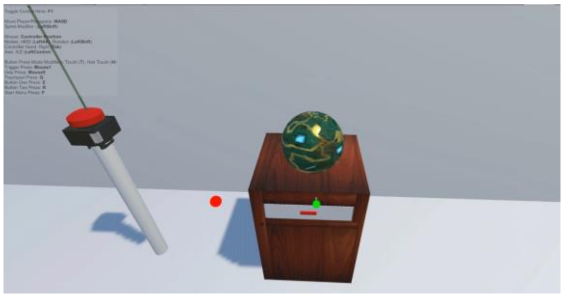
Gif 6. Spawn ball



Gif 7. Animated Shader

Placing the ball onto the big button to unlock the chest

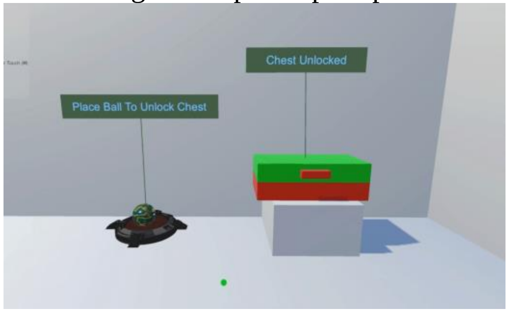
After the ball is spawn, the player can grab the ball by holding down the VR controller grip. The player will have to hold on to the ball and walk/teleport near to the big button and carefully place the ball on the button. The button will animate to appear being held down by the weight of the ball and this will unlock the chest. If the ball rolled off, the button will relax back to original position and the chest will stay lock. We originally planned for separate uses for lava and circuit ball that will produce different effects when used on the same big button.



Gif 8. Grab Ball Move

Grabbing and Lifting up the lit of the weapon chest

After placing the ball weighting down on the button, the chest will unlock and allow the user to grab the handle using the grip button and lift the hinged lid up to flip it open.



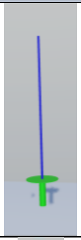
Gif 9. Chest Flip Open

Originally the chest contains a key to be used to unlock the drawer of the wooden desk and the player can drag open a drawer to obtain the weapons(gun).

Gif 10. Open Drawer

Grabbing Weapons from inside chest and using them

The chest contains 3 different weapons:

Gun	
Rapier	
Lightsaber	

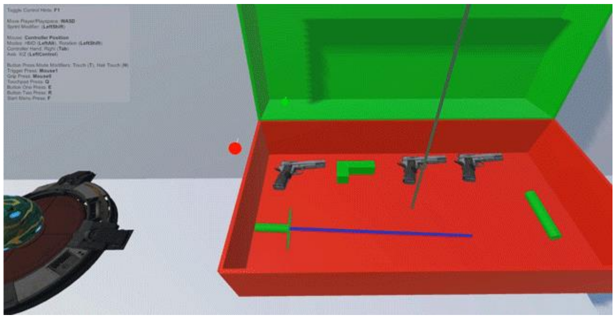
The player can equip any weapons using the VR controller grip button. The weapon has a fixed position and rotation for the attached point and it is used attached to the virtual hand.

Killing the spider that blocks the exit

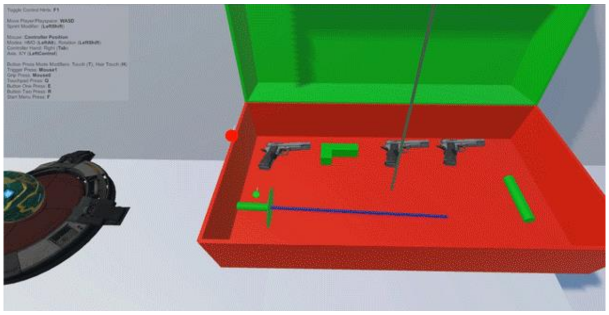
Next is to kill the spider.

- The player can equip the gun and press the trigger button to shoot a bullet to kill the spider.
- The player can equip the rapier and move the VR controller to thrust or swing the blade to hit the spider to kill it.
- The player can also use the light saber, press the trigger to extend the beam to hit the spider to kill it. (not implemented)

The spider uses two animations, one idle animation and one death animation. The spider plays the idle animation in a loop by default. Upon being collided with a weapon (bullet or blade) the spider will play the death animation and will despawn after animation ends. The spider also uses a special dissolve shader to give the effect of the spider disappearing gradually instead of disappearing instantly from the game.

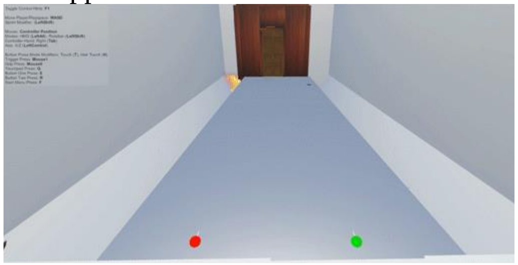

Toggle Control Hints: F1
Move Player/Playspace: WASD
Sprint Modifier: (LeftShift)
Mouse: Controller Position
Modes: HMD (LeftAlt), Rotation (LeftShift)
Controller Hand: Right (Tab)
Axis: XZ (LeftControl)
Button Press Mode Modifiers: Touch (T), Hair Touch (H)
Trigger Press: Mouse1
Grip Press: Mouse0
Touchpad Press: Q
Button One Press: E
Button Two Press: R
Start Menu Press: F

Gif 11. Gun Kill



Gif 12. Rapier Kill

After the spider disappear, a bridge will spawn, and the player can walk across to reach the exit. The bridge uses the same disappearing effect to appear.



Gif 13. Walk Across Bridge Open door

The wolf is similar to the spider but its unkillable and falls over when hit with a weapon instead of disappearing. The puzzle mechanic for this is not yet implemented.

The spider was originally a maze level filled with spider enemies that will move randomly using Unity navMesh nodes. The player has a gun and needs to navigate (restricted to walk only) to find the exit. The spider has all the different types of walking animation as well as an attack animation to kill the player and the player will have to restart. This is replaced with Level 2

Climbing a ladder (VRTK)

The last step is to walk across the bridge and climb the ladder to go to the next level. The player needs to coordinately use the left and right VR controllers and use the grip button to grab the ladder handle and pull themselves up to the next level.

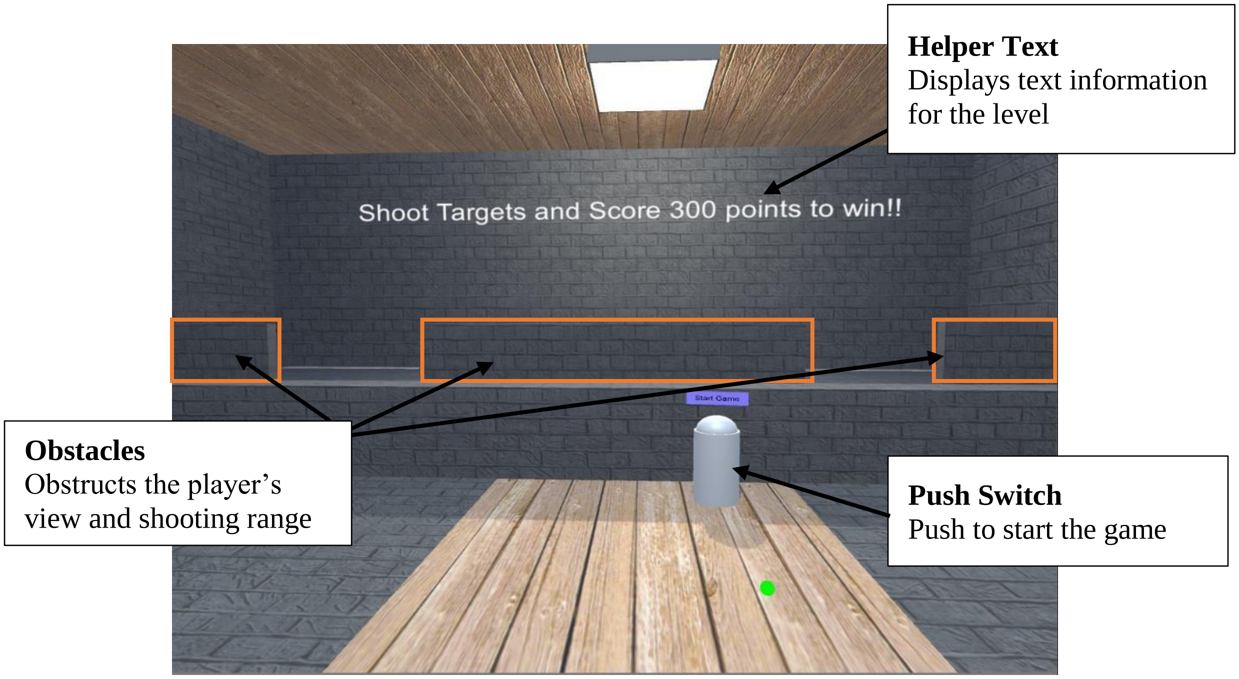
Gif 14. Climb Ladder

We initially plan to have a portal instead of a ladder. The portal will teleport the player to next level and once the player reached the next level on the other side of the portal, the portal will close, and the player will no be able to return to previous level. There will also be a gate that destroy any object that the user was grabbing to prevent the player from using the previous objects for the next level.

Level 2

Layout

Ascending the stairs from Level 1 will lead the player into Level 2 of the game. Level 2 consists of a shooting gallery in which the player has to clear before proceeding. Refer to Figure 1 for the overall layout of the level.



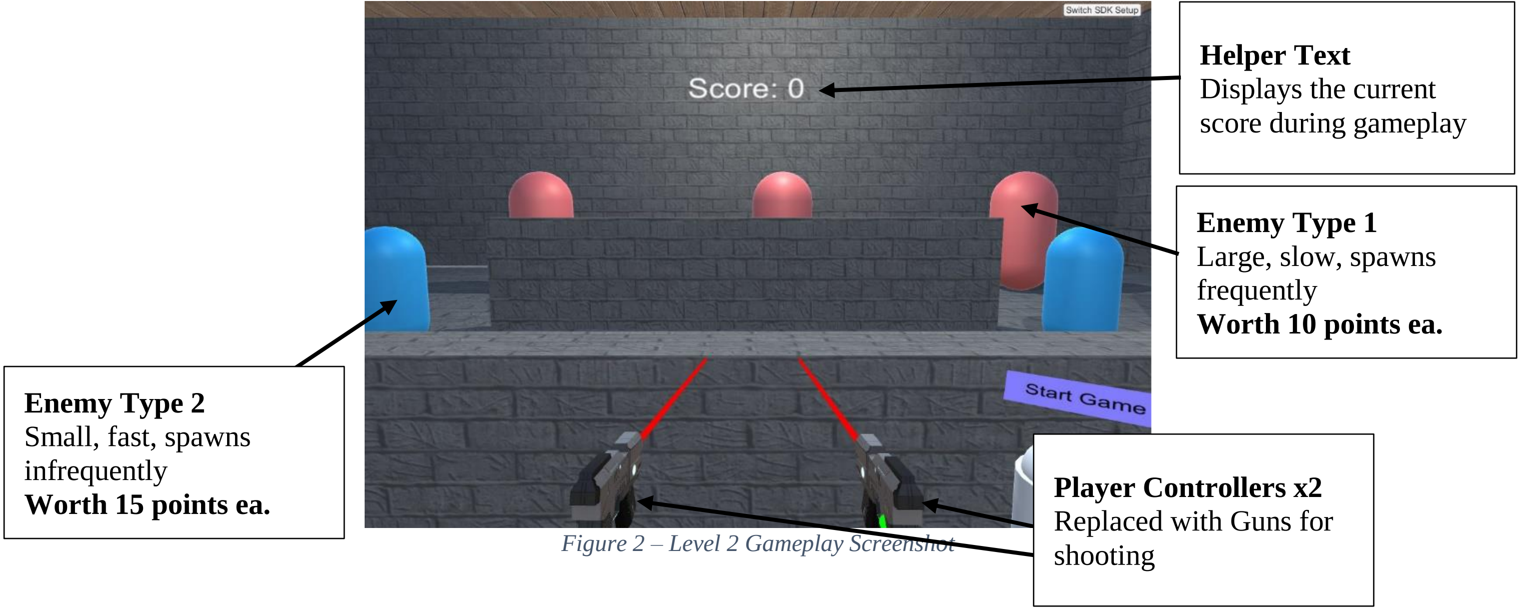
Helper Text
Displays text information for the level

Obstacles
Obstructs the player's view and shooting range

Push Switch
Push to start the game

Figure 1 - Level 2 Layout

To proceed, the player has to obtain 300 points from the shooting gallery as stated by the Help Text on the wall behind the shooting gallery. To begin the level, the player must first approach and push the push switch at the front of the shooting gallery. Once the switch is pushed, the player's controllers will be replaced with pistols and enemies will start spawning and moving on the shooting gallery. Players are provided laser pointers on the gun tip to help guide in shooting.



Helper Text
Displays the current score during gameplay

Enemy Type 1
Large, slow, spawns frequently
Worth 10 points ea.

Enemy Type 2
Small, fast, spawns infrequently
Worth 15 points ea.

Player Controllers x2
Replaced with Guns for shooting

Figure 2 – Level 2 Gameplay Screenshot

The player is able to move and teleport around the area of the shooting gallery as they please, but they are unable to teleport or move inside the shooting gallery itself.

Each enemy is programmed to spawn on the right of the shooting gallery and will move towards the left, where it will then disappear. With guns in hand, the player will be able shoot at the enemy to earn points to proceed on with the next stage. Refer to Figure <?> for a breakdown on the scoring.
Additional control functionalities are given to the player while playing the level. The controls are:

- Trigger: Shoot Gun
- Button 2: Switch Weapon Type

During the level, the player is able to swap out a weapon type with another to achieve a better effect at shooting at the enemies. The overall aesthetic of the gun does not change when switching weapon type. The weapon catalog available is as follows:

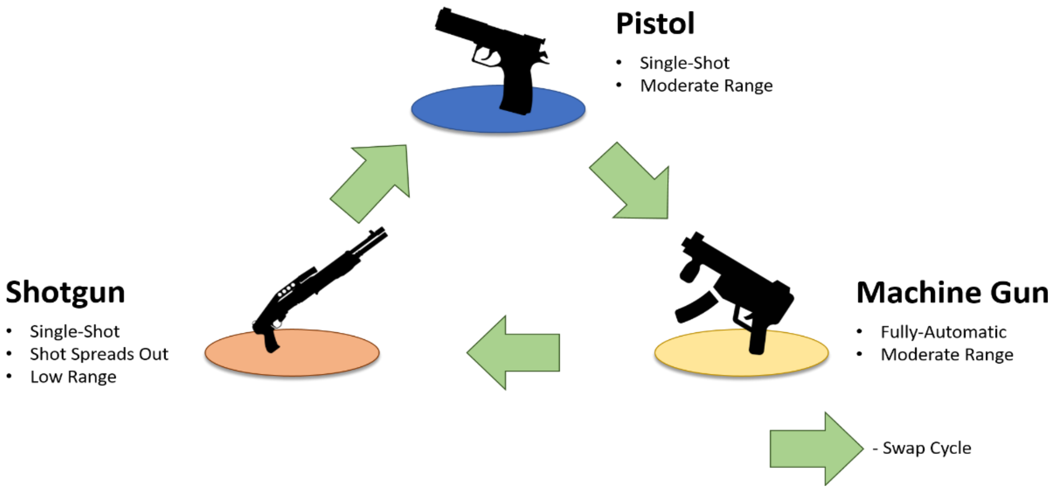
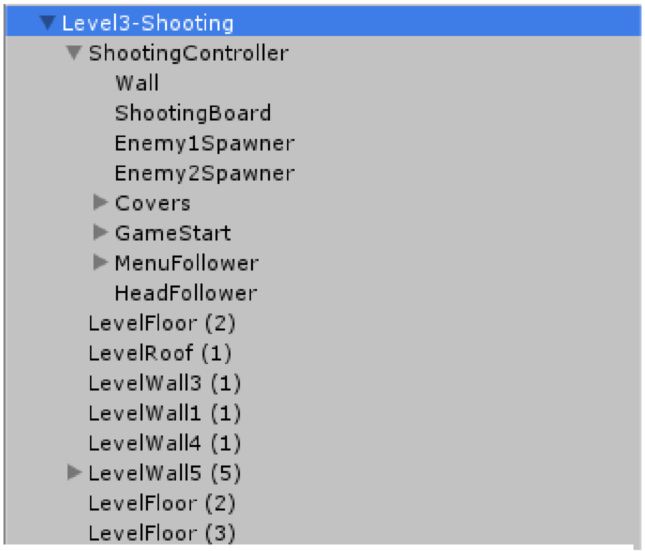


Figure 3 - Weapon Swap Cycle

Each weapon type is unique in its own way and it is up to player preference on which weapon type they prefer to use. Once the player has achieved 300 points in the shooting gallery, the level ends and the guns are removed from the player controllers. The player may choose to play the level should they wish to by pressing the push switch again.

Implementation

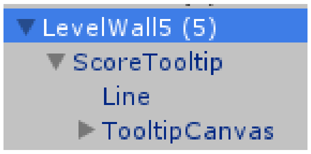
A main level controller Game Object, ShootingController, is implemented to encapsulate every component of the level to allow easy porting and ensures modulation between the other levels of the game. The ShootingController Game Object contains a ShootingController script to handle the core mechanics of the level, mainly scoring and when the game ends. Each Game Object related to the game level will either have the ShootingController Game Object as a variable or vice versa for interactions between the Game Objects when the game level is running. The ShootingController script contains a variable, gameStart, which indicates whether the game level has started or has it ended. Each of the game level's Game Object will reference this variable to monitor the state of the game level.



```
▼ Level3-Shooting
  ▼ ShootingController
      Wall
      ShootingBoard
      Enemy1Spawner
      Enemy2Spawner
    ▶ Covers
    ▶ GameStart
    ▶ MenuFollower
      HeadFollower
    LevelFloor (2)
    LevelRoof (1)
    LevelWall3 (1)
    LevelWall1 (1)
    LevelWall4 (1)
  ▶ LevelWall5 (5)
    LevelFloor (2)
    LevelFloor (3)
```

Figure 4 - Shooting Gallery Level Hierarchy

Each LevelFloor, LevelRoof and LevelWalls (From Figure <?>) denotes the four walls, the floor and the ceiling of the level. LevelWall5, the back wall behind the shooting gallery, contains a VRTK tooltip containing a Canvas to be used to display the Help Text. The ScoreTooltip object is stored as a variable in the ShootingController script to alter the Helper Text at points during the game.



```
▼ LevelWall5 (5)
  ▼ ScoreTooltip
      Line
    ▶ TooltipCanvas
```

Figure 5 - LevelWall5 Layout

The EnemySpawner1 and EnemySpawner2 Game Objects each have an Enemy Spawner script that generates an Enemy1 and Enemy2 prefab respectively at a set interval. The Enemy prefabs are created from the base capsule Game Object from Unity. Each of it has a controller script that generates movement for the Game Object when it is created and a distance limit that when reached, will destroy the Game Object to signify that the Enemy has reached the end of the shooting gallery. The prefabs also have a rigidbody to be used for collision detection during shooting.

Once the game level ends, every Enemy prefab created is destroyed immediately and the Spawner Game Object stops spawning any further Enemy prefabs.

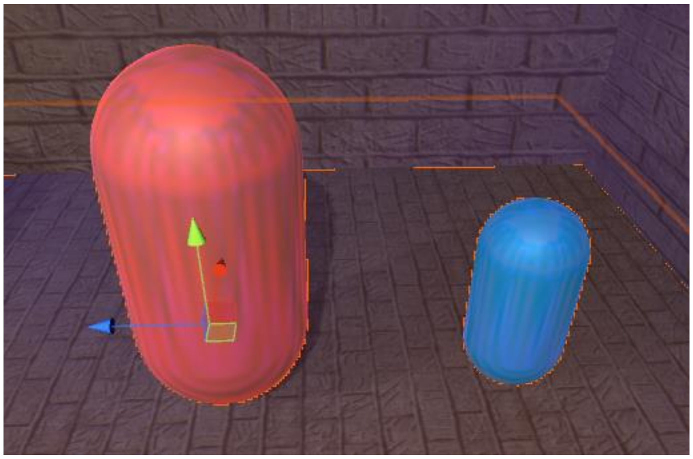


Figure 6 - Enemy1 (Left) & Enemy2 (Right) prefabs

Shooting Mechanic

Shooting is primarily handled by the pistol prefab, SciFiHandGun, that is initialized and attached to the player's controllers when the game level starts. For the pistol prefab to be attached to the player's controllers, it must first be able to interact with the player's controllers. To achieve this, the pistol prefab's controller extends VRTK_InteractableObject to mark it as an interactable object. Then, a VRTK_FixedJointGrabAttach script is added to the prefab to attach it to the player's controller when it is grabbed. But if the pistol is attached as it is, it will not be pointing forward but instead downwards as there is no point of attachment defined on the pistol. To remedy this, a snapPoint Game Object is added to the pistol prefab to define its orientation and displacement when attached. VRTK_ChildOfControllerGrabAttach script is added with the snapPoint Game Object to handle the attachment point and direction.

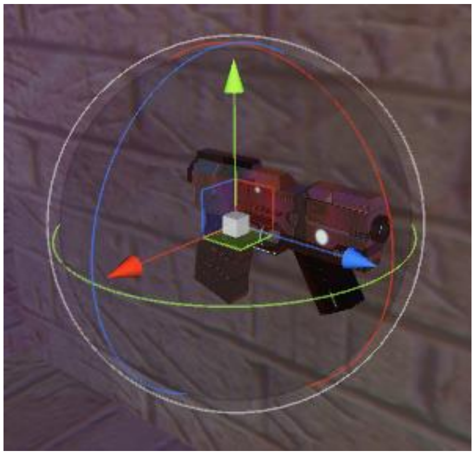


Figure 7 - Screen Capture of the pistol prefab with the snapPoint Game Object

A Muzzle Game Object is added to the front of the pistol prefab to indicate where the shot will be fired from. When the player fires a shot with the Trigger button, a Ray will be created and shot forward from the Muzzle with a set distance limit. Any collisions occurring with the Ray and a rigidbody will be registered and if the rigidbody collided is of an Enemy prefab, it will register as a hit to the Enemy, destroying the

Enemy prefab and awarding the player points. Particle effects like muzzle flare on the Muzzle and bullet impact on the ray’s point of collision to the ground or wall are created for visual feedback that a shot is fired. Similarly, a Haptic Pulse is also played on the controller to generate physical feedback. A laser prefab is added to the muzzle to serve as a marker to where the player will be shooting at with the pistol prefab.

Once the game level ends, the pistol prefabs are simply unattached from the controllers and destroyed respectively.

Weapon Switching

The pistol prefab fire different types of shots based on the which firing module it is at. If the firing module is pistol, a single ray is created with each single Trigger button press. For machine gun module, rays are created constantly while the Trigger button is held till it is released. For the shotgun module, rays are created similarly to the pistol module with each Trigger button press, but instead of a single ray, multiple rays are projected at once instead, each with a different direction generated using Random.Range.

To allow the player to switch between each firing module, a SwitchGunController script is added to the left and right controller scripts of VRTK to attach the change weapon method from the pistol prefab to the event when Button2 on the controller is pressed.