SpellBound

A MIXED REALITY GAME

Abhinav Joshi 3035203857

CHEUNG CASADO, Sheng Leung 3035123863

COMP 4801

15st April 2018

1. Summary/Abstract

Virtual reality as a platform has taken off recently with the advent of industry leaders like HTC, Oculus by Facebook and even apple trying to secure the ever-spreading market. Additionally, the gaming and entertainment industry is expected to gain the most with the spread of the new Technology.

However, it is the mixed reality sector which has been overshadowed. Projects and games rarely incorporate the physical world with the virtual world. With them either being wholly in virtual space like Oculus rift or augmenting reality like the HoloLens. To have total immersion in the Virtual space while having tactical and physically feedback in the real world creates a possibility to total immersion.

We, the development team (the Team) aim to develop Spellbound, a Multiplayer virtual reality game with aspects of and role-playing game (RPG) and gameplay of a first-person- shooter (FPS). Furthermore, to create a sense of physical engagement and the Team has also developed haptic vests to showcase how integrating physical feedback can result in an immersive gaming experience.

2. Acknowledgment

I would like to express my deepest appreciation to those who have provided me with opportunity to work on this project. I would like to specially express my gratitude to my supervisor Dr. CHOI without her I would never have thought of undertaking this project.

Table of Contents

1. Summary/Abstract 1
2. Acknowledgment 2
3. List of Figures 6
4. List of Tables 7
5. Abbreviations 7
6. Introduction 8
   6.1 Overview 8
   6.2 Background 8
   6.3 Objective and Scope 9
   6.4 Current Status of Mixed Reality 9
   6.5 Deliverables 10
   6.6 Outline of the report 10
7. Related Works 10
   7.1 A Vox eclipse and dino destroyer 10
   7.2 Altspace VR 11
   7.3 Haptic Vest development 11
8. Requirements 12
   8.1 Hardware 12
      8.1.1 Leap motion 12
      8.1.2 VR Headset (HTC Vive) 12
      8.1.3 High performance computers 12
      8.1.4 Arduino boards (Haptic vest controllers) 12
   8.2 Software required 13
      8.2.1 UNITY (Game engine) 13
      8.2.1 Leap motion API (Orion) 13

8.2.2 HTC Vive API 13

8.2.3 Arduino Software (IDE) Arduino Web Editor 13

9. Methodology 14

9.1 Software Development Practice 14

9.2 System setup 14

10. Game Development 15

10.1 About the Game 15

10.2 Objective 15

10.3 Game Flow & Mechanics 16

10.4 Evaluation Phase in detail 17

10.5 Performance Phase 18

10.6 Resolution Phase 20

10.7 Lane System (Player movement) 20

10.8 Leap motion controller 21

10.9 HUd design 22

11. Development and Testing 22

11.1 3d assets 23

Projectiles and Particles 23

Player Prefab (Knight) 25

Game Map 26

Performance Phase 26

11.2 Lane Raycast 27

11.3 SteamVR integration 27

Steam VR 27

IK (Inverse Kinematic) 27

11.4 Networking 28

11.5 RPC (Remote procedure call) 28
11.6 Custom Vest 28
11.7 Arduino (Part of the Custom vest) 29
11.8 HtC vve 30
12. Future Goals 30
Oculus support 30
Server browser 30
Smoother animations 30
Multiple Maps 30
Character Selection 31
Steam Direct 31
13. Limitations/Difficulties Encountered 31
Inter-device compatibility 31
Mitigation 31
Lack of experience 31
Mitigation 32
Time Constraint 32
Mitigation 32
Hand Rigging using leap 32
Mitigation 32
Character Mesh 32
Mitigation 33
Leap Motion Hardware 33
Mitigation 33
Uniduino 33
Mitigation 33

Motor Power 33
Mitigation 34
Networking 34
Mitigation 34
Animation with IK 34
Mitigation 34
14. Conclusion 35
15. References 36
APPENDIX 39

3. List of Figures

Figure 9.9-1 Architecture Overview of System for Spellbound. Under motion capture the three devices (from top to bottom) are: Microsoft Kinect, Leap Motion, HTC Vive. 14
Figure 10-1Performance Phase Top down View 18
Figure 10-2 Attack Gesture and Projectile (Encircled in Red) 19
Figure 10-3 Shield prefab on Players Hand 19
Figure 10-4 Grab gesture. Player on the right using Grab to drain the other players Shield Ammo 20
Figure 10-5 The lanes in the Game (un-highlighted state) 21
Figure 10-6 Leap Hands in action. (encircled in red the current position and rotation of players right hand) 21
Figure 10-7 Player HUD UI 22
Figure 11-1 Attack projectile 23
Figure 11-2 Grab particle effect 24

Figure 11-3 Shield Prefab 24

Figure 11-4 Leap Motion Rigging 25

Figure 11-5 Knight prefab with IK Rigging 25

Figure 11-6 Game Map 26

Figure 11-7 Teleport effect 26

Figure 11-8 Lane Highlight 27

Figure 11-9 Compromised Position (if all positions are mapped) 28

Figure 11-10 Compromised Rotation (if all rotations are mapped) 28

Figure 11-11 Custom Vest design 29

Figure 11-12 Arduino with vibrating motors being tested 29

Figure 13-1 Latex like skin in character model hands 33

4. List of Tables

Table 11.1 **Evaluation outcome……18**

5. Abbreviations

VR: Virtual Reality

MR: Mixed Reality

UI: User interface

HUD: Heads Up display

Transform: The rotation and position of the object

6. Introduction

6.1 OVERVIEW

The game features Mixed reality, a culmination of Virtual Reality with real time Physical feedback, to help the User immerse onto the Virtual Space. This is achieved with an in-House implementation of a haptic vest to provide the players a sense of touch and Leap motion to provide real time free hand movement in the virtual space.

6.2 BACKGROUND

There have been Multiple market projections forecasting the rise of virtual, augmented, and mixed reality (VAMR) applications onto the next multibillion-dollar industry by the next decade [1, 2, 3]. With the advent of AR applications such as Pokémon Go! [4]and a plethora of consumer grade hardware such as the Oculus rift, HTC Vive, PSVR or Microsoft HoloLens being adopted at a high pace [5,6]. The demand for highly immersive and quality games is rising. The communities created by users, can attest to such claims with users demanding more immersion as the line between reality and the virtual space slowly fades, exemplifying the need of more VAMR apps. Virtual reality (VR) is defined as a computer-generated 3D environment, which can be interacted by a person, and is expected to become a household name in the likes of the entertainment and gaming industry. VR devices are the first front to this union appealing heavily to the senses of the vision to the users, with Haptic devices providing further higher degree of simulation by creating the perception of touch and feedback, enhancing the VR experience [7].

The research efforts in creating a visual, auditory, kinesthetic, and haptic VR technology started off as early as the late 80s to early 90s, with fully commercial Hardware available to the users. However, it was not until the early 2010s that widespread public attention came to VR when the likes of Google, HTC, Samsung, Sony, Facebook, Microsoft etc. started to invest in this technology [8]. VR headsets quickly gained traction and became the hot topic in the gaming industry as the next evolution of games. The $2 billion acquisition of Oculus VR by Facebook in 2014[8], the cheap and easily availability to VR by the introduction of Google Cardboard and the announcement of the Sony PlayStation VR created a whole new industry waiting

for new development [9]. At the same time, development of several haptic devices came into place, but despite the technology outburst, these never gained similar traction [13].

The need to provide a simple solution of an immersive game with already proven technologies is required. Standardizing and providing simple libraries and compatibility between these haptic devices is required [13].

6.3 OBJECTIVE AND SCOPE

The main objective of the project is to develop, SpellBound a multiplayer virtual reality game that contains strategic and FPS elements and at the same time integrating the game with other technologies like-

- Leap Motion to provide an immersive experience by implementing game interactions through hand gesture recognition and movement in the virtual world.
- In-House haptic vests capable of providing basic tactile feedback depending on a game event (e.g. receiving damage, or blocking a shot, low HP etc.)
- HTC Vive to emerge the player into a virtual world where his movement is reflected onto the game.

6.4 CURRENT STATUS OF MIXED REALITY

Mixed reality had a very early start the Aura Interactor being one of the earlier haptic gaming devices being available in 1994 [11], but it suffered from lack of applications and development. However, with the rise of new technologies, user interest is increasing with big companies like Apple, HTC and google investing in VR. However, what they lack currently in is standardized hardware and software support. focus. Moreover, the majority of available games utilizing haptic feedback have little to no support for VR games, and the rest mostly still under development. The focus of the project is to provide a game changing mixed reality solution utilizing various proven technologies such as HTC vive and Leap motion along with DIY haptic vest with simple APIs.

6.5 DELIVERABLES

The final product we plan to deliver is a Mixed reality game which uses the HTC Vive, Leap motion and requires the windows platform. Along with the game, 2 haptic vests developed by the Team are to be delivered. The following is a detail list of deliverables

- An FPS/RPG combat game with simple character models and map along with the implementation of core gameplay mechanics
- Network support for 2 players.
- Full support for VR using HTC Vive headset
- Hand recognition in the virtual world for interaction using the Leap motion controller.
- 2 haptic suit prototype which are integrated with the game.

6.6 OUTLINE OF THE REPORT

The remaining paper describes the works that inspired the project and the various technologies used (section 7). This includes briefly describing the Hardware and software and the reasoning behind their usage (section 8). Then, it briefly describes the gameplay mechanics of our project along with some basic UI (section 10). Later, the paper explains how we plan to implement it and Lastly, all the difficulties encountered and future of the project(section 11-14).

7. Related Works

7.1 A VOX ECLIPSE AND DINO DESTROYER

Vox eclipse is a VR game requiring the use of an Oculus Rift and a Rift-mounted Leap Motion Controller to enable the user to use his/her hands freely in the virtual world and interact with items using hand gestures such as picking up, pinch to zoom etc. Users use their hands to pick up rocks and shoot at dinosaurs. Reviews of the game praise its unique approach and the usage of leap motion as a controller to offer players a unique experience. The only thing lacking in the game it the sense of touch / feedback while picking up rocks, making the game feel less immersive and gimmicky.

7.2 ALTSPACE VR

AltspaceVR is a multiplayer open world VR game. It offers players from around the world interactive games and social hub to hang around. It utilizes a VR headset and leap motion controller to interact with the environment. It is a free to play game with a huge community of players. It has support for both HTC Vive and leap motion. This has provided an excellent opportunity to study the behavior of people in VR using the leap controller while at the same time look at how the developers have implemented their design with the Leap controller. Especially the usage of gestures which is one of the most sensitive areas when choosing the leap as your main input source.

7.3 HAPTIC VEST DEVELOPMENT

A haptic vest or suit is a piece of equipment worn by the user to provide him/her a feeling of tactile feedback when he/she interacts with the environment. This creates a sense of immersion with the user when used along Auditory and visual simulation.

The idea of a sensory feedback system is not new. The Aura Interactor was one of the first instances of earlier haptic gaming devices. Conceived by Larry Shultz and launched in 1994, it was first of its kind to use bass sound waves to deliver vibrations providing the user a sense of touch [11]. However, this idea didn’t catch on due to lack of software and proper game titles to support it. But with the advent of VR and the outreach to users with the use of platforms such as Kickstarter many new companies have ventured into the haptic vest technology.

In 2017, Woojer a startup, started manufacturing a vest with better implementation using the latest technology along with a SDK for developers to provide a experience similar to what the Aura Interactor hoped to achieved [15]. A major competitor to Woojer is KOR-FX, another startup launched in 2014 from a successful Kickstarter campaign. The KOR-FX vest already has tie ins with games such as Resident Evil 7 and is available to consumers [16]. However, what it lacks in is proper game support and a User base.

8. Requirements

8.1 HARDWARE

8.1.1 Leap motion

The leap motion is a small device designed to be placed on either a physical desktop facing upwards or mounted onto a VR headset. It utilizes two monochromatic infra-red cameras and multiple LEDS to reconstruct 3D objects in its path to a virtual 3D object. It can be used as a highly specialized device to capture users hand gestures precisely [15] .

8.1.2 VR Headset (HTC Vive)

One of the key equipment required is the correct VR headset and it is imperative to choose the perfect headset for maximum compatibility of our game. The two main contenders were the Oculus Rift, or the HTC Vive. While both provide excellent VR experience and are similar in specifications, the deciding factor was spatial tracking. The Vive uses a lighthouse laser tracking system whereas the Oculus uses a Constellation based tracking system. The issue with Constellation system arises as the maximum space tracked by the camera is eight by eight meters. In comparison, the lighthouse laser system is capable of a maximum tracking space of 15 by 15 feet [14].

8.1.3 High performance computers

Virtual reality offers an enjoyable experience but at the cost of computational power. To provide the graphical capabilities like high frame rates, HD textures and high resolution a powerful computer is required. To power the GPU at least a Nvidia 970M a top tier graphic card is required along with a Dual core i5 or higher processor.

8.1.4 Arduino boards (Haptic vest controllers)

To provide the vest with a controller The Arduino Mega best suits the requirements. It is a single-board microcontroller which will interact with Unity using the USB serial commands. Along with the microcontroller will be the actuator motors responsible for firing up to give the user the sensation of touch as haptic feedback. The initial focus for the development process was to create mechanically induced haptic sensations and

later the possibility of including more transducers, actuators, or vibrators was be entertained.

8.2 SOFTWARE REQUIRED

8.2.1 UNITY (Game engine)

Unity is a free game engine that supports development of a variety of game types from 2D platformers to VR. It also supports a huge library of VR plug-ins. It provides a highly versatile and flexible environment for game development. From 3D Animations to spatial-audio. Unity provides an array of tools for developers to use as per their requirement. Moreover, with a strong community of developers and great product documentation, Unity offers the best solution.

The Team is also more familiar with Unity development, its documentation, and its community more than other well-reputed game engines such as Unreal.

8.2.1 Leap motion API (Orion)

Leap provides robust support for its product the Leap Motion. In early February 2016, Leap Motion released a major software update dubbed Orion, the updated software was designed specifically for precise hand tracking in virtual reality making it a valuable peripheral for the MR game [24]. This software is easily integrated with unity with a simple plugin.

8.2.2 HTC Vive API

HTC Vive utilizes the steam VR plugin with the unity engine to streamline the transition between a normal game and a VR game [23].

8.2.3 Arduino Software (IDE) Arduino Web Editor

Arduino provides a IDE and a web editor to simplify the interaction between the Arduino boards and computer. Much of the low-level programing for the haptic vest is based on this software [21].

9. Methodology

9.1 SOFTWARE DEVELOPMENT PRACTICE

Due to the limited timeframe and just two developers, the most suitable development practice is to use the Agile methodology [22], which is feasible for flexible requirements and rapid changes in the development process or environment. Therefore, the Team decided to organize development into small milestones which can be achieved on a weekly basis and separating the workload to prevent mismatch in version control. These weekly iterations will be evaluated at the beginning and end of every week.

9.2 SYSTEM SETUP

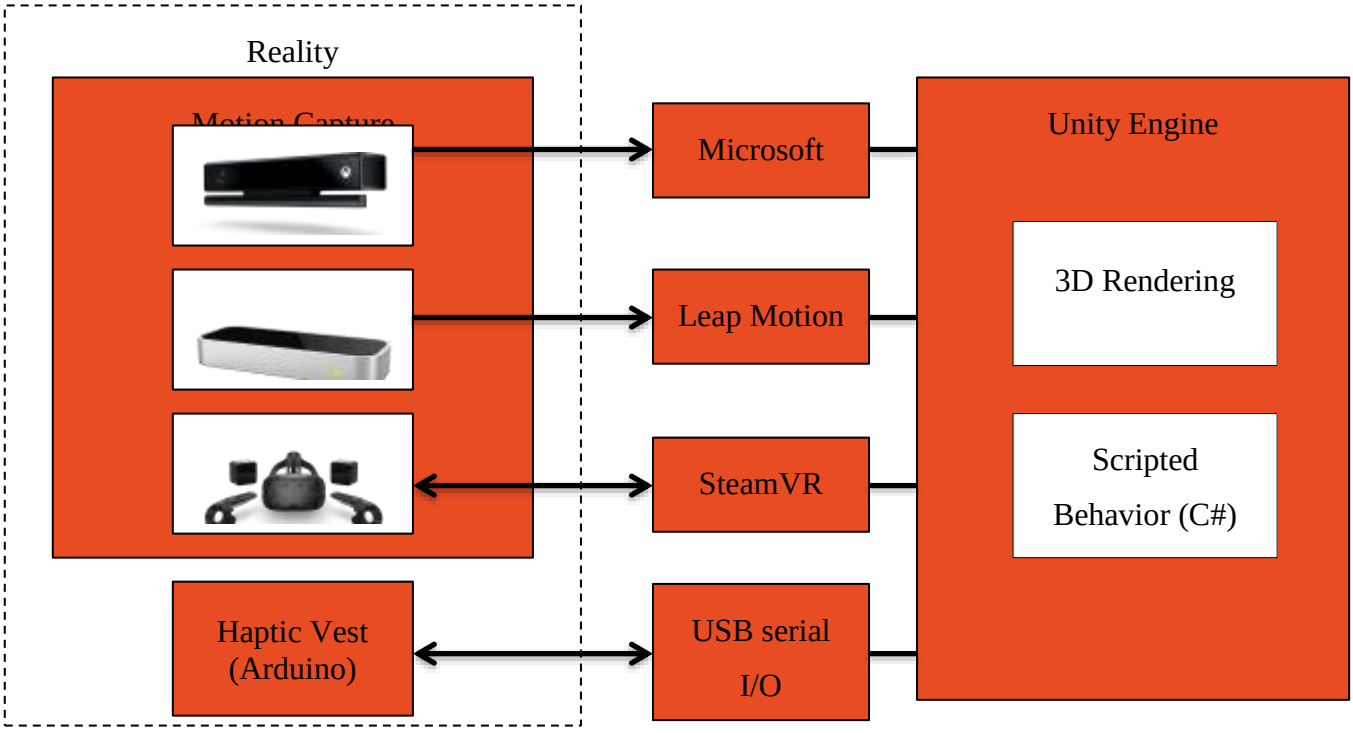


Figure 9.9-1 ***Architecture Overview of System for* Spellbound. *Under motion capture the three devices (from top to bottom) are: Microsoft Kinect, Leap Motion, HTC Vive.***

The above figure (see Figure 9.9-1) outlines the general idea of the project where:

- The lines to the Unity Engine block connects to the relevant SDK, libraries, and plug-ins enabling the devices to communicate with the engine
- The arrows represent the transfer of data as per the direction. Hardware input is translated by the relative API solutions to enable the Unity engine to understand the various input, while Hardware output is translated from Unity by their corresponding APIs/plugin.
    - The Leap Motion provides the translation of hand gestures to accurately determine the users hand movement.
    - The Kinect provides a skeleton structure of the users making their Body movements appear in the virtual space. (Future aspect)
    - The HTC Vive setup provides Visual and auditory information to the user while at the same time determines the spatial position and the orientation of the User.
    - The Arduino board on the haptic device awaits signals to turn on the actuator motor to simulate a tactile feedback.

10. Game Development

10.1 ABOUT THE GAME

SpellBound is Co-op mixed reality game with aspects of Both an RPG (Role playing game) and an FPS (First Person Shooter). It is a turn based game compromising of four phases (see section 10.3) with a section compromising of the player to perform on his action. Players have a variety of character classes and maps to choose from and can play with either an AI or Online.

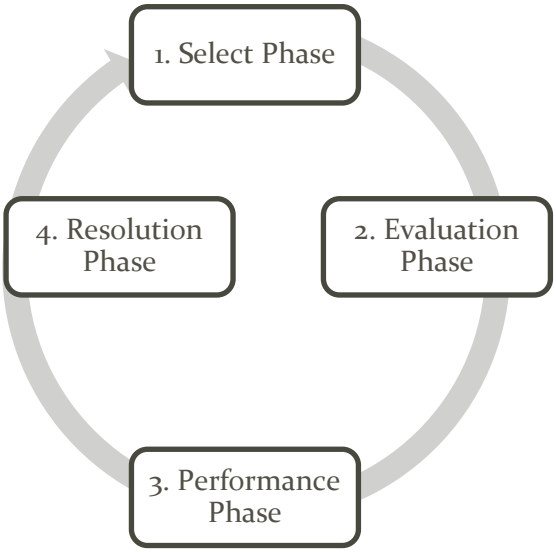
10.2 OBJECTIVE

The goal of the game is simple, to damage the other player as much as possible while at the same time trying to stay alive. The game ends when either of the player dies or the timer ends. The winner is either the player alive or in case the timer expires the player with the least amount of damage inflicted upon.

10.3 GAME FLOW & MECHANICS

To start the game the two players, join a lobby from the Main menu. Once successfully paired, they are placed on opposing sides of the map in the middle lane. The game then starts as depicted in Figure 10.1.

Figure 10.1 **The different phases in *Spellbound*.**



The game can be divided into 4 phases.

1. Select phase

   The user is prompted to choose between a variety of options mainly Attack, Defend(shield) Grab and Move.

2. Evaluation Phase (Fig 10.1)

   The user ends his/her turn and his move is evaluated based on Table 10.1. By the game

3. Performance Phase

   The user is prompted to partake in some action based on the move he chooses. Example if he chooses to attack he might be prompted to point and shoot his opponent. If he/she choose to move this will be the time to do so. This action is time bound

4. Resolution Phase

   The user is now shown the result of his action and his/her status like health, magic etc. are updated. The user then shifts back to the selection phase.

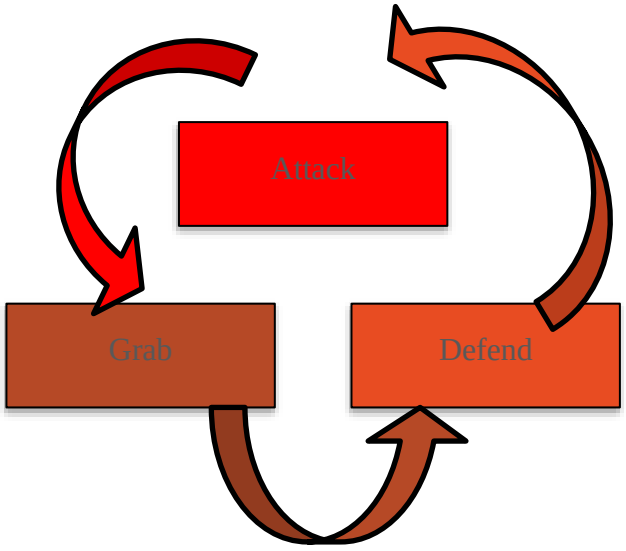


Figure 10.2 ***Rock-paper-scissors nature of the game mechanics in* Spellbound**

Table 1.1 Evaluation outcomes

USER 1	USER 2	SOLUTION
Attack	Shield	USER 1: - Receiving reflect damage USER 2: - No damage
Attack	Attack	USER 1: - Damage USER 2: - Damage
Attack	Grab	USER 1: - Safe USER 2: - Extra Damage
Attack	Move	USER 1: - Safe USER 2: - Movement -1 Correct Guess USER 1: damage Incorrect Guess USER 1: safe
Shield	Shield	USER 1: - Safe, energy deducted USER 2: - Safe, energy deducted
Shield	Grab	USER 1: - Energy deducted USER 2: - Safe, energy gained/absorbed
Shield	Move	USER 1: - Safe, energy deducted USER 2: - Safe, Movement -1
Grab	Grab	USER 1: - Energy deducted USER 2: - Energy deducted
Grab	Move	USER 1: - Safe USER 2: - Correct Guess USER 1: energy deducted Incorrect Guess USER 1: safe

*Grab: if energy depleted

10.4 EVALUATION PHASE IN DETAIL

The evaluation phase deals with resolving the choices taken by the players and presenting the player with the results in the performance phase. Table 1.1 defines the outcomes based on choices a player can take in detail. Column 1 and 2 define the choices made by player 1 and 2 respectively, while column 3 defines the net outcome of their choices

10.5 PERFORMANCE PHASE

After the players have selected their choices and the Game engine decides on the outcome (Evaluation phase), if the outcome requires player interaction (Example P1 Attacks P2 and selects the right lane). The players are transported to a 1 on 1 lane where each player can visually see the other players exact movements and gestures (see Figure 10-1). The players are free to move around in the confined lane and try to doge the other players attacks.

Figure 10-1Performance Phase Top down View

Here depending on the players choice in the Selection phase (see Section), the players powers are enabled. The players are the prompted to use their powers to either Damage or shield the other player. The performance phase has a set Timer for the players to interact with each other. When the Timer expires the players are teleported back to spawn. The following are brief descriptions of how the players interact based on their choices.

Attack

The player fires a flaming projectile using the Attack Gesture (see Figure 10-2). These projectiles on collision with the other player damages him/her and enables haptic feedback on the vest.

Figure 10-2 Attack Gesture and Projectile (Encircled in Red)

Defend

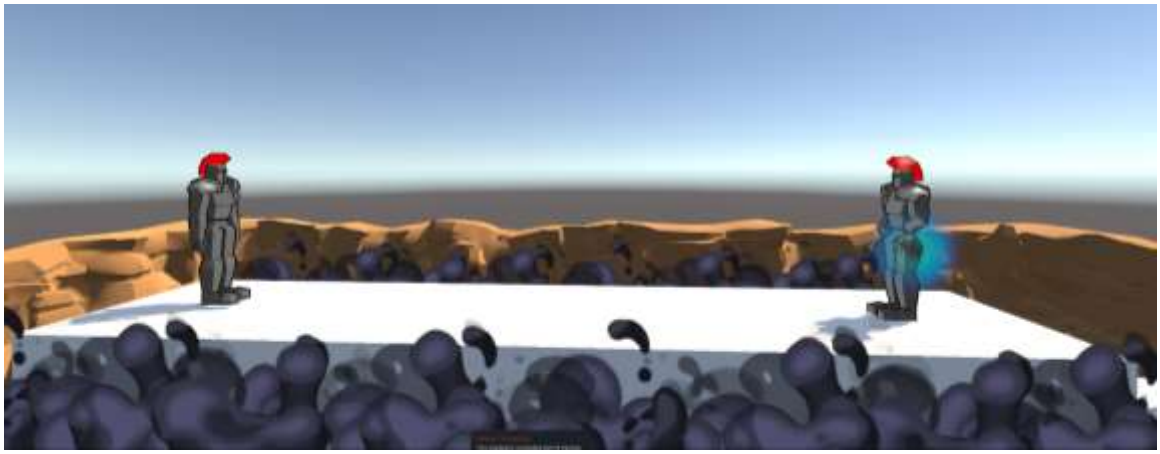


Figure 10-3 Shield prefab on Players Hand

The player is prompted to use the Shield gesture to create shields on his/her hands. This enables the player to block incoming Attack projectiles and prevent Damage from occurring. (see Figure 10-3)

Grab

The player uses the grab gesture to enable the grab effect. This effect drains the other player shield ammo if the other player chooses to shield and increase the users shield ammo in turn. (see Figure 10-4)

Figure 10-4 Grab gesture. Player on the right using Grab to drain the other players Shield Ammo

10.6 RESOLUTION PHASE

When the Performance Phase ends the players are teleported back to their spawns. Here they are updated with Information like Damage taken, ammo spent/ reduced etc. Selection Phase is started again (see section 10.3).

10.7 LANE SYSTEM (PLAYER MOVEMENT)

The movement of the player each turn is restricted by his adjacent blocks. If the Player chooses to move he/she can only move to the available place in white. This is done to restrict the number of places a user can move, increasing the pace of the game. This system might be improved upon if different characters classes are made available i.e. Some classes may enable the player to move more spaces, while some might restrict the movement further

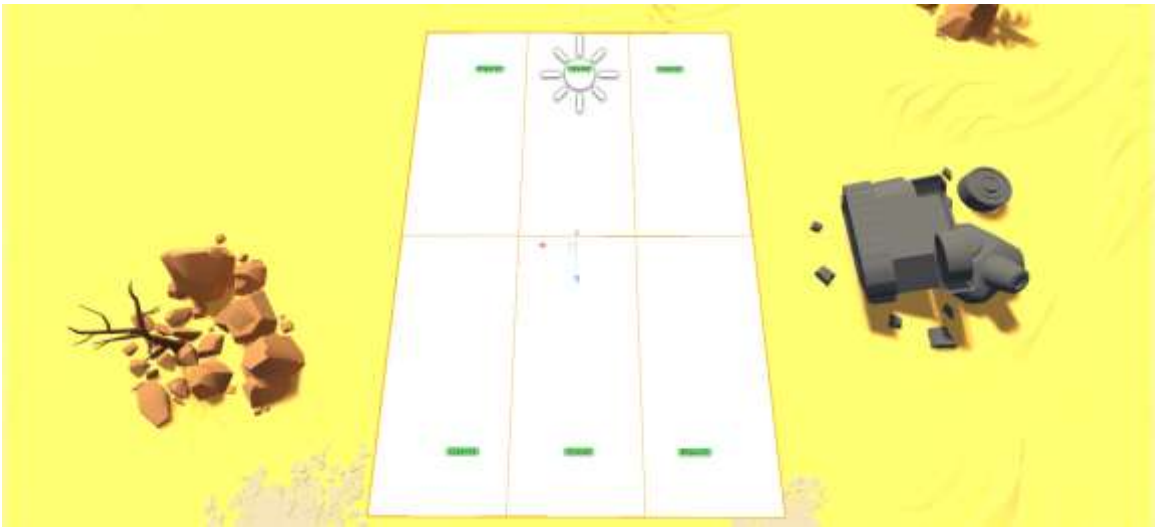


Figure 10-5 The lanes in the Game (un-highlighted state)

10.8 LEAP MOTION CONTROLLER

The player interacts with the environment and menu using his hands which are mapped out by the Leap motion controller (see figure 10-6). Precise hand and finger movement can be established. This is a major input source for the user as he/she will be prompted in the performance phase to create a gesture to represent the action such as to shield place your palm up.

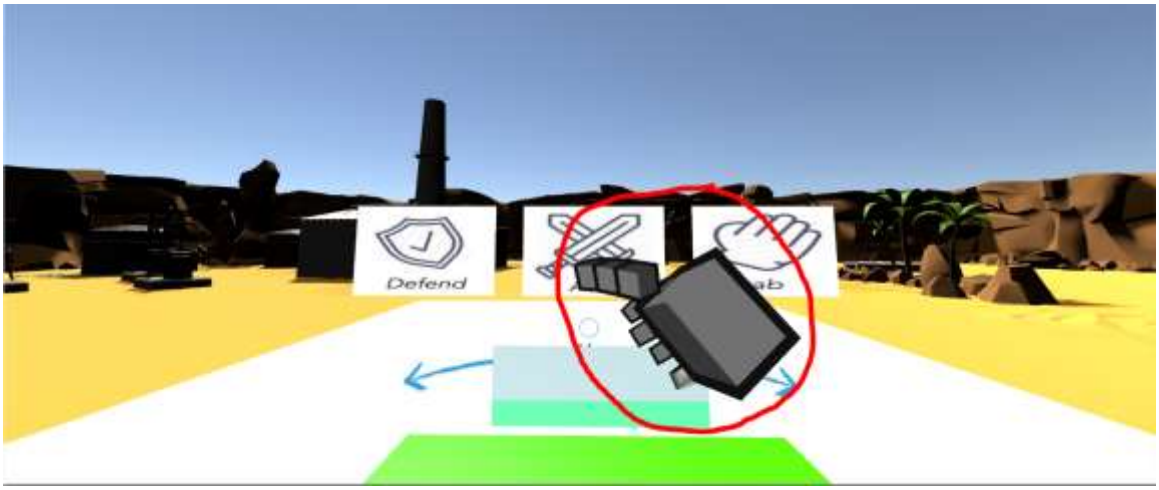


Figure 10-6 Leap Hands in action. (encircled in red the current position and rotation of players right hand)

10.9 HUD DESIGN

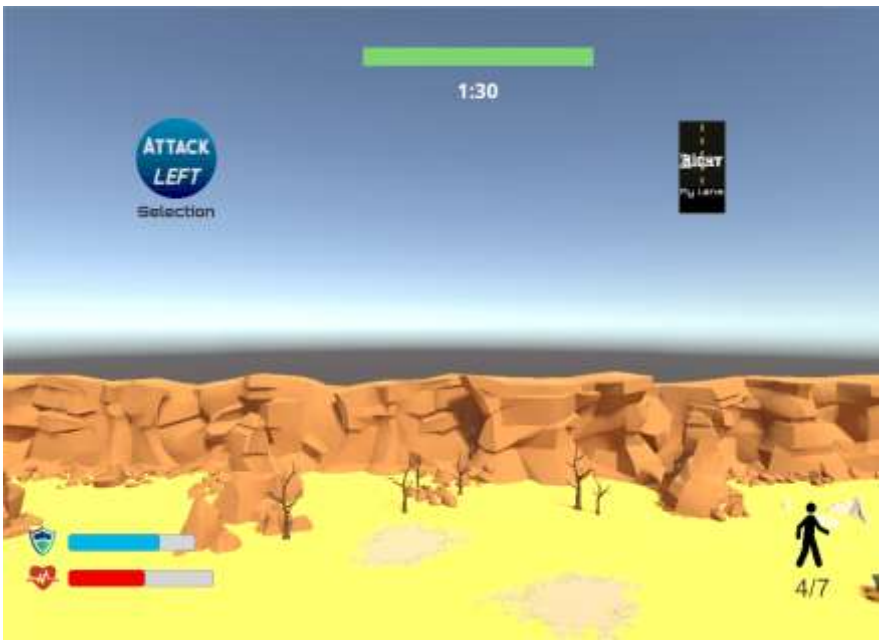


Figure 10-7 Player HUD UI

The player Heads up display or HUD is a screen overlay of useful information the player might require (see Figure 10-7). This information keeps on getting updated based on the choices of the player as well as per the instructions of the Game Engine. Below is a brief description on the different UI labels in the HUD.

- Top-Left: The choice made by the player in the selection phase
- Top-Center: The time left in the current phase
- Top-Right: The current lane the user is in (Will be depreciated later)
- Bottom-Left: The shield ammo (in blue) and the player Health meter (in red)
- Bottom-Right: The Move ammo left

11. Development and Testing

Extensive testing and development on various environments and hardware was done. Our findings for the following aspects of SpellBound are as follows

11.1 3D ASSETS

To develop 3D assets, various platforms like Unity and 3Ds max were used. As 3D max offers a more friendly and easy approach to work with and with its compatibility with unity, it was the optimal choice for developing 3D models. However, as Unity offers a vast library of Game specific assets and being the software developing environment, multiple game assets were used. Below are the list of assets created or used in the final game

Projectiles and Particles

Attack projectile

The attack projectile consists of a sphere collider used to detecting if the projectile hit any body part of the player (This is important to enable Haptic feedback on a particular body part). The firey effect is created using the Unity particle effect(see Figure 11-1). The sprets for the effect were imported from the asset store.

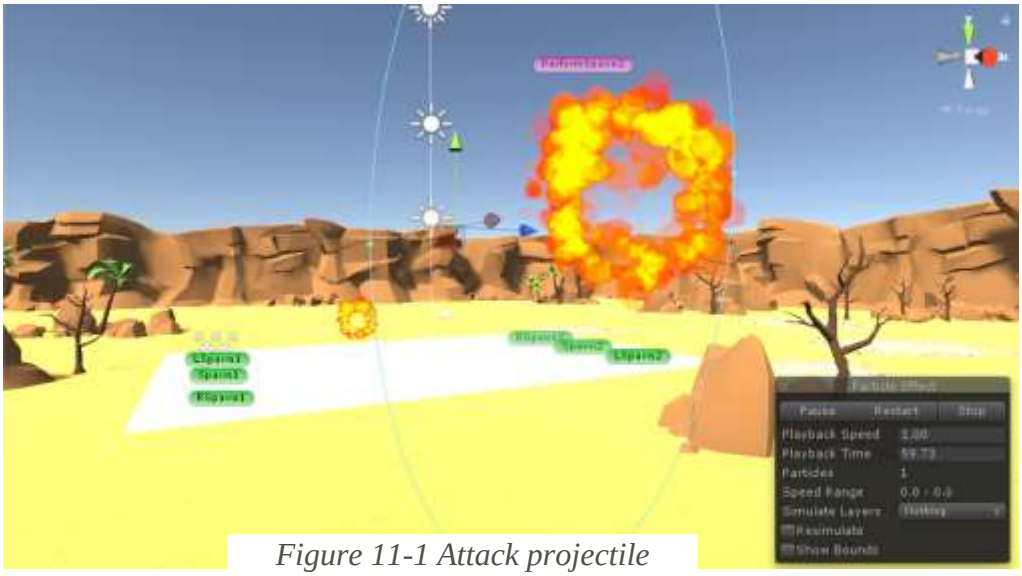


Figure 11-1 Attack projectile

Grab effect

The Grab effect was created with the help of the asset store and a custom script which enable the particles to have a set target to follow. This creates a drain like effect visible in the figure 10-4 in section 10.5. The particles originate from the player the

Grab is cast to and gather towards the hand that cast it. See figure 11-2

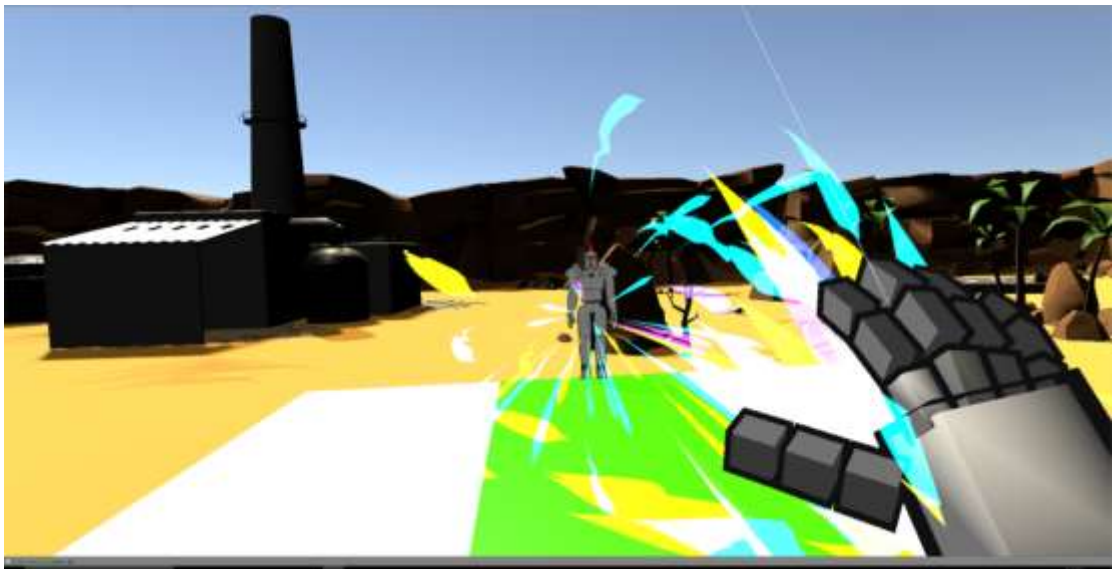


Figure 11-2 Grab particle effect

Defend shield

An in house developed Prefab. It has a collider to detect incoming projectiles and destroy them as well as a radiating effect which looks pleasant to the eye.

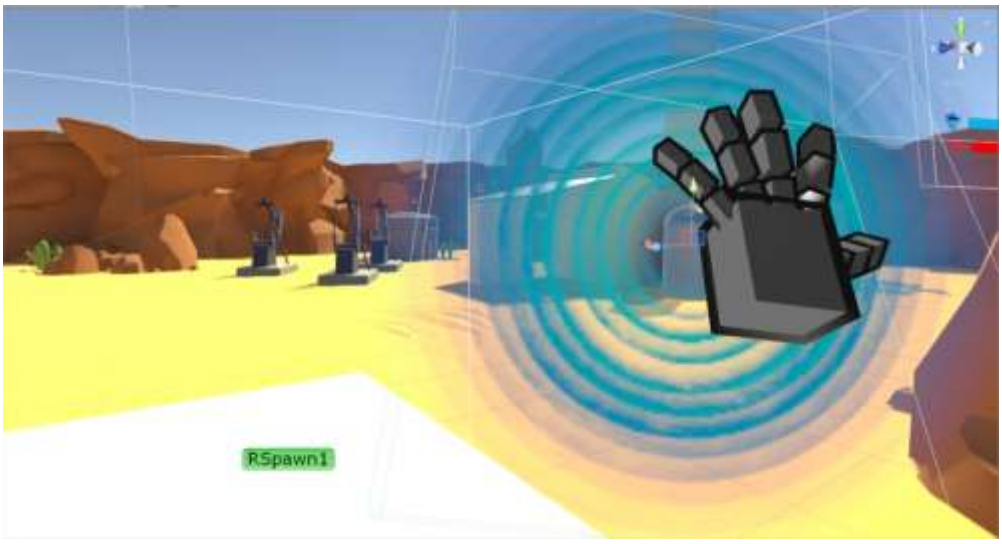


Figure 11-3 Shield Prefab

Player Prefab (Knight)

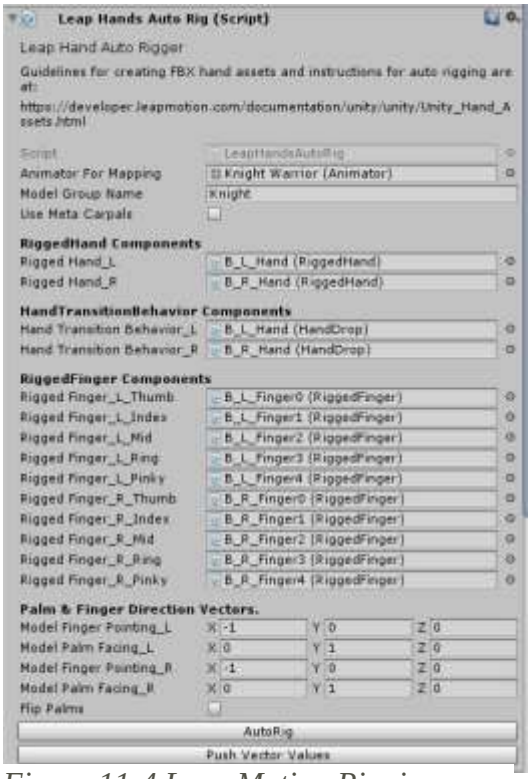


Figure 11-4 Leap Motion Rigging

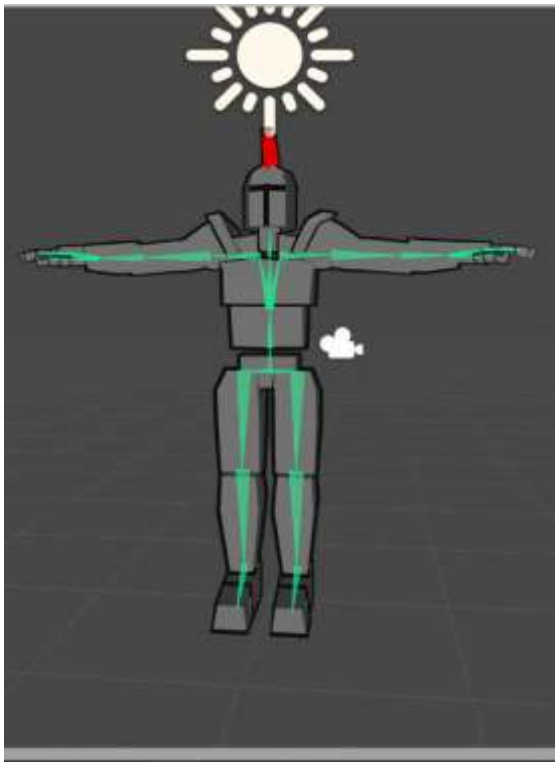


Figure 11-5 Knight prefab with IK Rigging

The player model uses a improved version of a Knight prefab from the asset store. The heavily modified prefab uses a Humanoid rig which enables the use of any animation supporting it (see Figure 11-5).

This also enables us to use the hands of the Knight and map them to the leap motion hands using the Leap motion Hand rig script (see Figure11-4).

This creates a humanoid body for the player to see in VR and experciene a fluid motion of hand and body movement.

Game Map

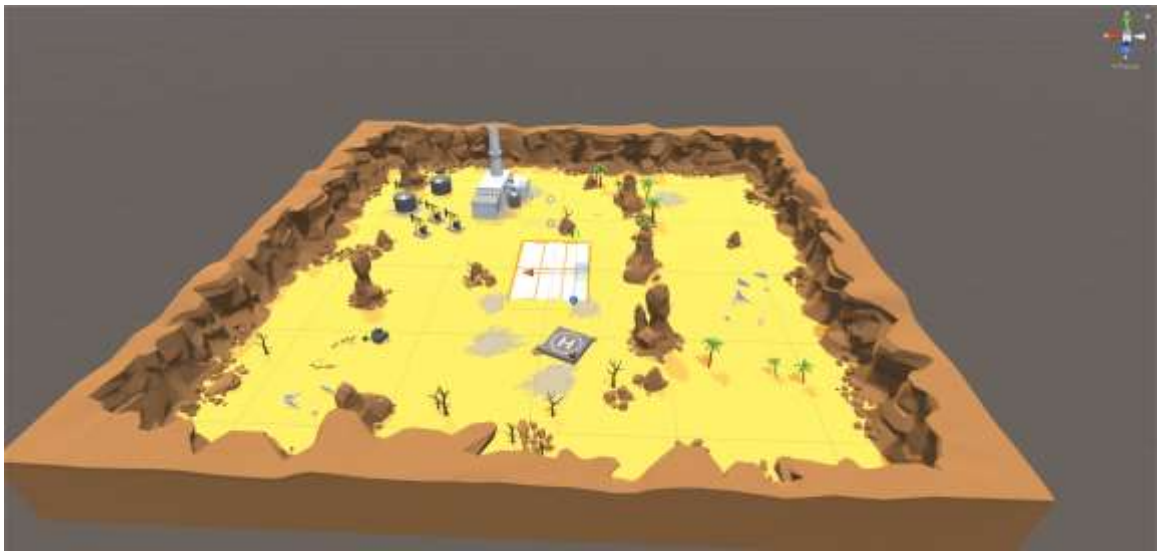


Figure 11-6 Game Map

The Game map was created from assets acquired from a unity Tutorial. However, the Map has been designed by the Team and assets heavily modified to suit out game and different 3D assets are added to the map.

Performance Phase

The performance phase When the player is teleported to the Performance Phase level an animated portal is rendered before his eyes. This smoothens the transition. As for the Performance level (see Figure 11-7 for a better idea) multiple particle systems are present to create a sense of urgency for the user.

Figure 11-7 Teleport effect

11.2 LANE RAYCAST

The player can always check and see his lane which is highlighted in Green (see Figure 11-8). This is done so the player can always keep in track of his/her movements. This effect was made possible by casting a ray from the legs of our player model and check if it collided with anything. If the conditions are met then the material mesh is changed from being plain white to the highlighted Green.

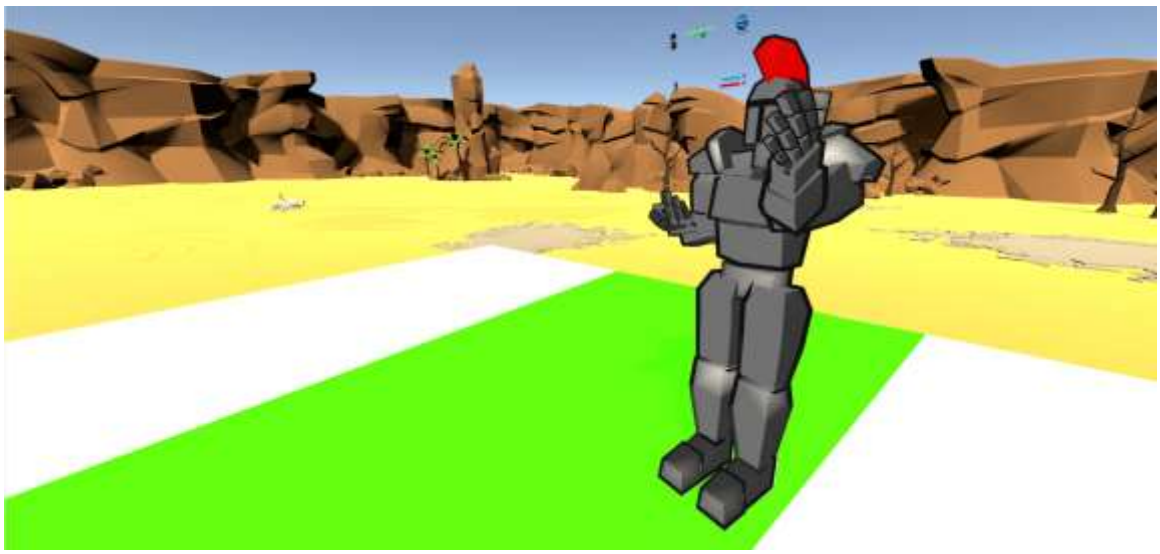


Figure 11-8 Lane Highlight

11.3 STEAMVR INTEGRATION

To enable VR in Unity the Team choose SteamVR library. The following are the implementation of the library as well as how the Knight model behavior.

Steam VR

This prefab is the building block of how unity interacts with VR. From getting the Transforms of the Headset and controllers to the mapping of the player in the Virtual world everything is accessed from here.

IK (Inverse Kinematic)

To Map the player headset rotation and position and to the Knight prefab we use a custom script involving Inverse Kinematic or IK which enables the Knight to be an embodiment of the player in VR creating the perfect immersion. The script utilizes the rotation of the headset only in the X- Z plane (parallel to the ground) and the position

of the headset only in the Z-axis and X-axis. This is done so the Knight prefab doesn't fall or be in a comprisable position as seen in figure 11-10 and figure 11-9.

Figure 11-10 Compromised Rotation (if all rotations are mapped)

Figure 11-9 Compromised Position (if all positions are mapped)

11.4 NETWORKING

Photon unity is used to send and receive information from each instance of the Game. To sync the transform of an object a PhotonView (Script) is required to be attached to that object. This script syncs the transform as well the object information which can be anything form collider triggers to rigid body information.

11.5 RPC (REMOTE PROCEDURE CALL)

To sync the variable in a script simply attaching a PhotonView is not possible as doing so creates a huge overhead and causes several issues. To do so a function overload must be called in the script called onPhotonSerialiseView (). This function is a basic serializer and de-serializer and should be dealt with carefully so only the Parent client calls serializer and the receiver client calls de-serializer.

11.6 CUSTOM VEST

The Vest design was finalized and completed using an Arduino Mega, Bread board and multiple vibrating motors. However, the final vest has only been created on a make-shift vest compromising of a simple jacket. The Arduino board has been installed in the vest with multiple Vibrating motors attached to key locations of the body (see figure 11-11).

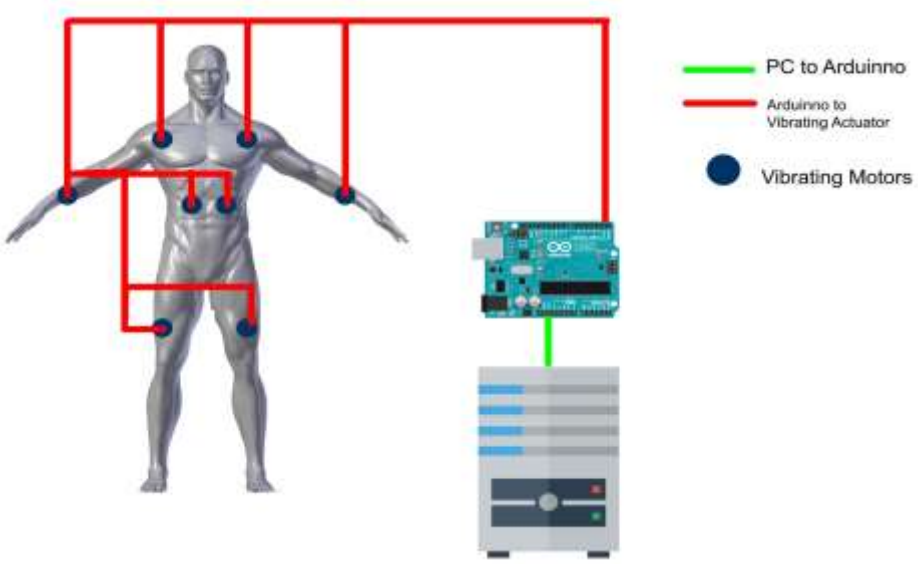


Figure 11-11 Custom Vest design.

11.7 ARDUINO (PART OF THE CUSTOM VEST)

The Arduino in use is an Arduino Mega 2560 with a custom program created for our game. This program is a serial USB reader which reads the input from the connected computer and matches it to the pre-defined functions present in the program. These functions activate the respective pins for a particular time (which can be set separately by calling another function) which in turn activates the Vibrating motors.

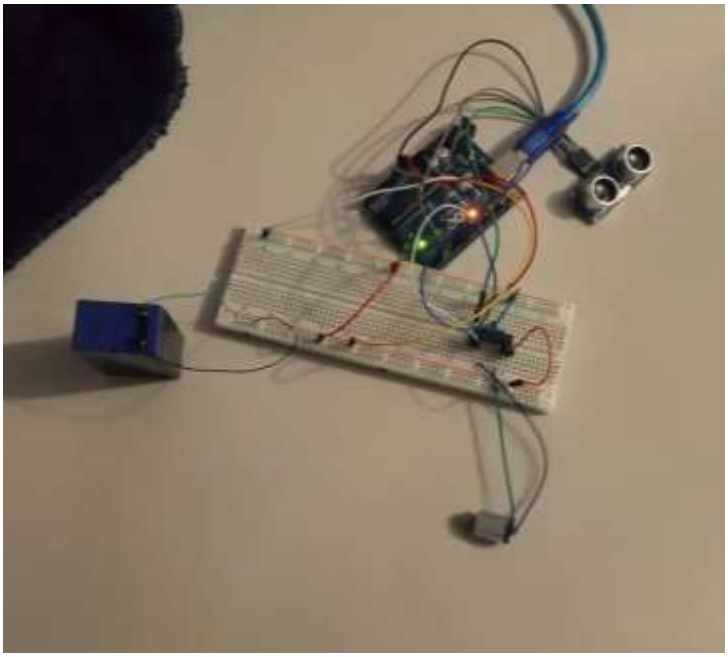


Figure 11-12 Arduino with vibrating motors being tested

For example, if the Computer sends a serialized string “LEFT ARM”, the Arduino will activate the respective pin for the Left arm.

11.8 HTC VVE

Testing with the HTC Vive has been sparse due to inaccessibility to the equipment till recently. However, the team has been using the VR simulator in unity to circumvent this hurdle. Gear VR is also being used as a substitute when working off location.

12. Future Goals

Due to time constraints the Team decided to add several features to be completed in the future. The following are a list of feature the Team plans to work on:

Oculus support

Currently only HTC Vive is supported as it utilizes the Lighthouse which is an important part of the game. By enabling movement using a keyboard or the oculus controller, a larger audience can be attracted (Currently 45% of VR users use the Oculus) [25]

Server browser

Currently connection with the server results in the player connecting to the available room. This room is either empty or has a single other player. In the future we plan to create a server/room browser so people can play with their friends directly.

Smoother animations

Much of the animation and effects added in the game are rough on the eyes although they server their basic purpose. Smoother transitions are required for emersion especially in VR hence the need for smoother animations

Multiple Maps

Initially multiple maps were planned to be in the game however due to time constraints and work on Higher priority issues, plans to do so were discarded.

Character Selection

Multiple Characters were planned initially and the game design and logic were built to support characters and abilities but as with the multiple map creation this was delayed and later discarded.

Steam Direct

An important future goal is to publish the game. Steam Direct provides a platform for game developers to publish their games directly on steam. This will enable our game to reach a massive audience.

13. Limitations/Difficulties Encountered

There are several difficulties which were encountered while working on Spellbound. Several goals had to be discarded or accommodate the worst case possible. Below is a list containing several problems we faced and their respective resolutions we tried to implement.

INTER-DEVICE COMPATIBILITY

As we planned to use a variety of devices each being made by a different developer, making them work together was challenging.

Mitigation

On further researching we found an extensible array of supporting documents for our different devices which will help integrate the hardware with the Unity software platform

LACK OF EXPERIENCE

The lack of experience in low level programming of the Arduino board along with working with hardware will be difficult. Moreover, the team did not have any prior knowledge on working with 3D assets.

Mitigation

Over the last few months working with 3D models has improved the familiarity with using 3D assets. On further research and testing on the Arduino board the team has gained some experience in Low level programming

TIME CONSTRAINT

To create the maps, textures, game objects, animations, sounds etc. requires a lot of time and resource both of which are few in quantity. A major challenge was to rapidly create these assets in time for the game release.

Mitigation

The team created a timeline for resource deployment and are strictly adhering to it. A moderate amount of time has also been allocated for any unforeseen circumstance or difficulties.

HAND RIGGING USING LEAP

To enable mapping of Leap motion hands to the ones of our character model created a major hurdle as the Team was inexperienced with Leap. We faced several issues from disorientation of hand in the virtual space to Collison between leap hands and character model hands.

Mitigation

The solution we found was to throw out the existing character model altogether and use a new Model which had a Humanoid Rig which enabled us to use the Leap auto rig script to map the hands

CHARACTER MESH

Even on rigging with the new Humanoid Rig the difficulty encountered was the latex like arms of our character model as visible in the figure 13-1

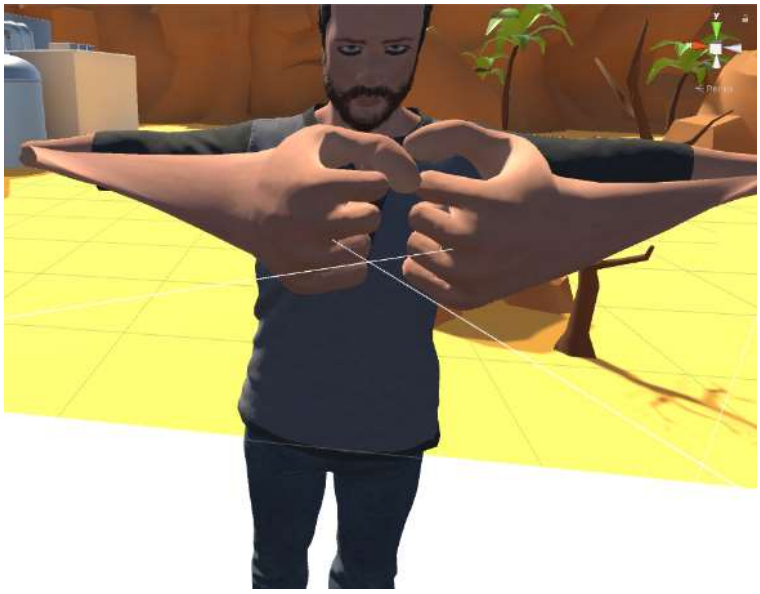


Figure 13-1 Latex like skin in character model hands

Mitigation

As our character model used a skin mesh to render the body we were unable to use it. Issue is still unresolved.

LEAP MOTION HARDWARE

The team had several issues while dealing with the leap motion hardware. From being unable to get data to having an inverse orientation of hands.

Mitigation

The only solution we could come up with was to test the leap before using.

UNIDUINO

Initially we planned to use Uniduino a plugin for unity to interact with the Arduino in our vest. However, the team had a difficult time work with Uniduino as its performance was not consistent.

Mitigation

The team decided to use native code support for the Arduino and use the USB serial connection to interact with unity.

MOTOR POWER

The vest has to power several motors and just using the Arduino was not possible as it could short circuit the board.

Mitigation

The Team contacted peers from the Electrical engineering department to get help with the power issues and came up with a better vest design which uses a L298 bridge and an external power source.

NETWORKING

Photon Unity provides a free cloud service to users to enable networking. However this has latency issues and has a bandwidth cap.

Mitigation

The team plans to deploy a private photon unity server when using the game.

ANIMATION WITH IK

Animating the character model while using Inverse Kinematics proved to cause several issues from leap hands to stop working to incoherent character movements

Mitigation

The character models had to be static (no animation currently).

14.Conclusion

SpellBound potentially offers a new and unique experience to users by providing an immersive and articulate environment. It fills the gap between the virtual world and the real world creating a Mixed reality for users to enjoy. By utilizing upcoming technologies, it makes way for the next leap in the gaming and entertainment. The game itself has its roots inspired by Classic games such as Skyrim, Final Fantasy etc. making it user friendly and at the same time enjoyable by all. The core mechanics create a challenging and fast pace game. Its rock, paper and scissor approach makes it easy to learn even when played the first time. Moreover, Unity engine provides the perfect software implementation by “Uniting” the various hardware elements. Although to play the game, some restrictions are present due to the hardware, by making the software open source in the future along with the details of the Haptic Vest as a DIY (do it yourself) kit, SpellBound hopes to create a simple and pleasant experience for the user. In the future, several new maps and characters will be added along with adaptive environments.

15. References

1. Chen, L. (2017). China's Virtual Reality Market Will Be Worth $8.5 Billion and Everyone Wants a Piece. [online] Bloomberg.com. Available at: https://www.bloomberg.com/news/features/2016-05-15/china-s-virtual-reality-market-will-be-worth-8-5-billion-and-everyone-wants-a-piece [Accessed 18 Oct. 2017].

2. Parker, L. (2017). Virtual Reality Is a Disappointment? Not in the World of Video Gamers. [online] Nytimes.com. Available at: https://www.nytimes.com/2017/06/22/technology/personaltech/virtual-reality-video-games.html [Accessed 1 Oct. 2017].

3. Statista. (2017). VR software market size worldwide 2016-2020 | Statistic. [online] Available at: https://www.statista.com/statistics/550474/virtual-reality-software-market-size-worldwide/ [Accessed 1 Oct. 2017].

4. Desk, W. and Unruh, J. (2017). Pokemon Go Fever: A look at the app that everyone is playing. [online] WGN-TV. Available at: http://wgntv.com/2016/07/14/pokemon-go-fever-a-look-at-the-app-that-everyone-is-playing/ [Accessed 10 April. 2018].

5. Lang, B. (2016). HTC Vive Headset Nearing 100,000 Sales - Road to VR. [online] Road to VR. Available at: https://www.roadtovr.com/htc-vive-sales-figures-data-100000-steamspy-data/ [Accessed 10 April. 2018].

6. Gothard, P. (2017). Microsoft HoloLens: Firm admits sales figures are in the 'thousands' | TheINQUIRER. [online] http://www.theinquirer.net. Available at: https://www.theinquirer.net/inquirer/news/3003380/microsoft-hololens-firm-admits-sales-figures-are-in-the-thousands [Accessed 10 April. 2018].

7. Lackey, S. and Shumaker, R. (2016). Virtual, Augmented and Mixed Reality. In: HCI International. Toronto: Springer, pp.235, 257, 485-486, 623-632, 654, 693, 725-726, 732-733.

8. Virtual Reality Society. (2017). History Of Virtual Reality - Virtual Reality Society. [online] Available at: https://www.vrs.org.uk/virtual-reality/history.html [Accessed 10 April. 2018].

9. iRoboticist. (2010). Tactile Gaming Vest (TGV). [online] Available at: https://iroboticist.com/2010/03/26/tgv/ [Accessed 10 April. 2018].

10. Chen, H. (1994). Electronic vest adds a chest full of thrills to video games. [online] tribunedigital-baltimoresun. Available at: http://articles.baltimoresun.com/1994-08-27/features/1994239088_1_interactor-aura-systems-vest [Accessed 10 April. 2018].

11. Woojer - Born To Feel. (n.d.). Vest™ - The #1 Haptic Gaming Vest for immersive games & VR experience. [online] Available at: http://www.woojer.com/vest/ [Accessed 1 Oct. 2017].

12. Korfx.com. (n.d.). KOR-FX 4DFX Haptic Gaming Vest. [online] Available at: http://korfx.com/ [Accessed 18 Oct. 2017].

13. Cava, M. (2016). Virtual reality's promise, risk loom large for health researchers. [online] USA TODAY. Available at: https://www.usatoday.com/story/tech/news/2016/02/02/virtual-reality-promise-and-concerns-both-loom-large-researchers/79360096/ [Accessed 1 Oct. 2017].

14. Digitaltrends.com. (2017). Oculus Rift vs. HTC Vive | Spec Comparison | Digital Trends. [online] Available at: https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/ [Accessed 18 Oct. 2017].

15. Developer.leapmotion.com. (n.d.). Gestures — Leap Motion C# SDK v2.3 documentation. [online] Available at: https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html [Accessed 18 Oct. 2017].

16. Msdn.microsoft.com. (n.d.). What's New in the October 2014 Kinect for Windows version 2.0 SDK. [online] Available at: https://msdn.microsoft.com/en-us/library/dn782041.aspx [Accessed 20th Oct. 2017].

17. Arduino.cc. (n.d.). Arduino - FAQ. [online] Available at: https://www.arduino.cc/en/Main/FAQ#toc13 [Accessed 20th Oct. 2017].

18. Uniduino.com. (2015). Getting Started » Uniduino. [online] Available at: http://www.uniduino.com/kb/getting-started/ [Accessed 20th Oct. 2017].

19. Technologies, U. (2017). Unity - Manual: VR overview. [online] Docs.unity3d.com. Available at: https://docs.unity3d.com/Manual/VROverview.html [Accessed 20th Oct. 2017].

20 Microsoft's Windows Mixed Reality (Oct 17, 2017) : everything you need to know - Tom Warren . [online]. Available at: https://www.theverge.com/2017/10/17/16487936/ [Accessed 20th Oct. 2017].

21 Arduino IDE and setup (2017): [online] Available at https://www.arduino.cc/en/Main/Software [Accessed 20th Oct. 2017].

22 Agile meathodology(2017) :[online] Available at: https://www.versionone.com/agile-101/ [Accessed 20th Oct. 2017].

23 HTC vive steam VR plugin: [online] Available at: https://www.assetstore.unity3d.com/en/#!/content/32647 [Accessed 20th Oct. 2017].

24 Orion Leap motion: [online] Available at: https://www.vrfocus.com/2016/02/leap-motion-announces-orion-for-faster-more-accurate-vr-hand-tracking/ [Accessed 20th Oct. 2017].

25 Oculus Rift Surpasses HTC Vive:[online] Scott Hayden Available at: - https://www.roadtovr.com/oculus-rift-takes-lead-htc-vive-steam-majority-market-share [Accessed Mar 2, 2018]

APPENDIX

Table 1.1 Use Cases for Resolution of Performance Phase

Case	Player 1 (P1)	**Result**		
	Player 2 (P2)		P2 correct lane	P2 incorrect lane
1	Attack	P1 correct	*Damage reflected* to P1 P2 shield energy decreased	*Damage reflected* to P1 P2 shield energy decreased
	Defend	P1 incorrect	P1 prompted incorrect P2 shield energy decreased	P2 shield energy decreased
2	Attack	P1 correct	P2 *receives extra damage*	P2 *receives extra damage*
	Grab	P1 incorrect	P1 *slightly damaged* P2 *absorbs P1 shield energy*	P1 prompted on miss P2 prompted on miss
3	Attack	P1 correct	P1 & P2 *damaged*	P2 *damaged*
	Attack	P1 incorrect	P1 *damaged*	P1 prompted on miss P2 prompted on miss
4	Attack	P1 correct	P2 *receives extra damage* P2 move ammo decreased	P2 *receives extra damage* P2 move ammo decreased
	Move	P1 incorrect	P2 move ammo decreased	P2 move ammo decreased
5	Defend	~~***P1 correct***~~	P1 shield energy decreased P2 shield energy decreased	P1 shield energy decreased P2 shield energy decreased
	Defend	~~***P1 incorrect***~~	P1 shield energy decreased P2 shield energy decreased	P1 shield energy decreased P2 shield energy decreased
6	Defend	~~***P1 correct***~~	P2 *absorbs P1 shield energy* P1 *slightly damaged*	P1 shield energy decreased
	Grab	~~***P1 incorrect***~~	P2 *absorbs P1 shield energy* P1 *slightly damaged*	P1 shield energy decreased
7	Defend	~~***P1 correct***~~	P1 shield energy decreased P2 move ammo decreased	P1 shield energy decreased P2 move ammo decreased
	Move	~~***P1 incorrect***~~	P1 shield energy decreased P2 move ammo decreased	P1 shield energy decreased P2 move ammo decreased
8	Grab	P1 correct	P1 & P2 *slightly damaged*	P1 *absorbs P2 shield energy*
	Grab	P1 incorrect	P2 *absorbs P1 shield energy*	P1 prompted on miss P2 prompted on miss

9	Grab	P1 correct	P2 *slightly damaged* P1 *absorbs P2 shield energy* P2 move ammo decreased	P2 *slightly damaged* P1 *absorbs P2 shield energy* P2 move ammo decreased
	Move	P1 incorrect	P2 move ammo decreased	P2 move ammo decreased
10	Move	~~***P1 correct***~~	P1 move ammo decreased P2 move ammo decreased	P1 move ammo decreased P2 move ammo decreased
	Move	~~***P1 incorrect***~~	P1 move ammo decreased P2 move ammo decreased	P1 move ammo decreased P2 move ammo decreased