MULTI-PLAYER MULTI-DEVICE GAME IN VIRTUAL REALITY.

Interim Report – FYP (COMP4801)

Date: 01-Dec-2016

Name: Riddhi Kasliwal
UID: 2013516917
Supervisor: Dr. Loretta Choi
THE UNIVERSITY OF HONG KONG
ABSTRACT

This report describes the design, implementation and evaluation of a project which explores new ways of multiple user interaction for a Virtual Reality application via a multiplayer game. It attempts to overcome the limitation of Virtual Reality (VR) applications which conventionally allow only one user wearing the headset to interact with the system. The game application aims to achieve a multiplayer effect by allowing one player to use a VR headset and other players to connect to the game via their mobile devices. The game design helps the users to get immersive experience based on the device they use to play the game. Therefore mobile phone players would have different role in the game compared to the VR player. This project is inspired by a game application made by some Masters students under this project’s supervisor.

This report discusses the progress made on the first and second stage of the project – Game Design, Networking and some difficulties encountered with Player’s Movements which is the ongoing third stage of the project.
ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Loretta Choi for helping me with this project’s idea. I am grateful for help received from the master’s student, Laurence Wong in setting up the project. I am thankful to Tina, another master’s student, for giving me the permission to use the models developed by her. Lastly, I would like to show my sincere gratitude to Professor Ken Ho for guiding me through the writing of this report.
TABLE OF CONTENTS

1. Introduction
   1.1 Background ......................................................... 5
   1.2 Related Works & Motivation ...................................... 5
   1.3 Objective .......................................................... 6
   1.4 Scope .............................................................. 6

2. Methodology
   2.1 Design ............................................................ 7
   2.2 Hardware & Software .............................................. 8
   2.3 Implementation .................................................. 9
   2.4 Testing ............................................................ 13

3. Schedule
   3.1 Timeline ........................................................... 15
   3.2 Current Progress .................................................. 16

4. Risks, Challenges & Mitigation ........................................ 17

5. Conclusion ................................................................... 17

6. References .................................................................... 18

LIST OF FIGURES AND TABLES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Architecture Diagram</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Vive Controller</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Camera Position</td>
<td>12</td>
</tr>
<tr>
<td>Table 1</td>
<td>Timeline</td>
<td>15</td>
</tr>
<tr>
<td>Table 2</td>
<td>Risks, Challenges &amp; Mitigation</td>
<td>17</td>
</tr>
</tbody>
</table>
ABREIVATIONS

VR – Virtual Reality.

CS – Computer Science

Vive – HTC Vive Virtual Reality Headset

AI – Artificial Intelligence

XP – Extreme Programming Methodology
1. INTRODUCTION

1.1 Background

Virtual Reality is an experience of completely immersing yourself into a virtual world. People frequently achieve/do things that they cannot imagine about doing in the real world using Virtual Reality for immersion experience. There are various levels of immersion for VR. For example, devices such as Google Cardboard which use mobile phone as a source of VR do not detect hand movements, hence the applications are limited to using mobile phone sensors. The other devices such as HTC Vive, Oculus Rift are accompanied with controllers to detect hand movements of the users. This significantly increases the variety of applications that can be made for Virtual Reality.

For the VR applications developed for such devices, the major limitation is that, only people wearing the headset can see the VR world. People surrounding the person wearing headset cannot get the feel of what the VR user is experiencing. Addressing this issue, this project aims to develop a multiplayer game which allows users to interact with the Virtual World using their mobile devices. So any number of people can be part of one experience. This experience will, however, depend on the type of device the user is using.

1.2 Related Work & Motivation

Currently, there are a few multiplayer applications for Virtual Reality such as Redout, Think Space [1]. Redout is a multiplayer gamewhich is desktop based with an optional VR headset – Vive or Oculus Rift support. It has online multiplayer compatibility which lets the players connect and play with anyone in the world. Think Space is only for VR headset – Vive and it also has online multiplayer compatibility. So users from anywhere in the world who have Vive, can connect to the game and interact in the Virtual World.

The limitations of games such as Redout is that it is not designed to give experience to the user based on the type of device the player uses. Therefore, the desktop and the VR player have the same role which leads to questioning the role of VR support for this game. For games such as
Think Space which are designed only for VR headsets, people without the headset cannot be part of the experience.

Considering these different type of multiplayer games, this project attempts to solve the issues mentioned above by introducing unique game design which assigns different roles to the users based on the device they are using. They can choose between VR headset and mobile phone to join the game and get a different experience according to the device used. This project will, as a result, explore the interaction of the camera positions and algorithms for different devices to give immersive experience accordingly.

1.3 Objective

The main objective of this project is to develop a game application which explores a new way for the mobile phone users to interact with a player wearing a VR headset in the same virtual experience. There will be different versions of the applications specifically designed for VR headset and for the mobile phone users to better their game experience. This project also aims to achieve networking between the different devices and implement a game artificial intelligence (AI) to control the game in favor of either the VR player or the mobile phone players to give immersive experience according to the device used by the player.

1.4 Scope

The features of this project for the first release will be:

- Different version for mobile and desktop (VR).
- 1-9 player compatibility (0-8 Mobile players, 1 VR player for one instance).
- Free movement around the game scene using Vive controllers for teleportation.
- Game AI to set traps for the players according to the player’s movements.
- Points, health potions, enemies, bombs, traps for competition among the players.
- Sound effects to indicate different actions carried out by the players.

However, this version would not include:

- Online multiplayer support.
- Customization of player avatar or the scene.
The final deliverable would be a working game application with immersive game design and a demo for people to experience the Virtual Reality game.

2. METHODOLOGY

This project will be developed using one of the agile practices, Extreme Programming (XP). This method follows a disciplined approach to deliver high quality software continuously in stages. This is achieved by frequent feedback, continuous testing and planning to deliver a working application every 1-3 weeks. Also the deliverable should be available in the next 5 months, therefore, this development methodology would give the optimal results.

The timeline for this project was set using XP and is described in Section 3 of this report. The first stage of the development cycle of this project is the game design. Before the final result of game design, there were several small results submitted to the supervisor for feedback. The third stage is the Players Movements. This paper describes these stages in the next few sub sections.

2.1 Design

This project is inspired by some post graduate student’s game application which explores networking between two HTC Vive headsets. The game is themed around Halloween where two players appear as giants in virtual environment. There is a wooden plank placed at a certain height over the city. The two players need to reach for the pumpkin which is placed in the middle of the plank. There is leap motion sensor attached over the headset which allows the game to track the hand gestures whereas the two controllers are attached to the feet to detect movements. Different hand gestures initiate different abilities which are used to defeat the other player. The first player to touch the pumpkin wins the game.

For this project, there will be only one VR player and the game will be extended to be compatible with multiple mobile devices. The mobile phone users will have different view of the game compared to the Vive user. The mobile phone users can see the VR players’ avatar on their phone with a third person view. Depending on the number of people who joined the game, the difficulty level would increase or decrease. The VR player acts as a witch and she needs to find
good pumpkins hidden by some minions, who are the mobile phone players, around the city. The witch cannot see all the minions at once but the minions can see the witch at all times since they would have a third person view of the city on their screens. The minions will have to predict the behavior of the witch and lay traps accordingly. There will be a game AI which would help the minions lay traps by suggesting the next move of the witch. The witch needs to find all the good pumpkins before the time runs out.

As stated in Section 1, this game is designed to give a satisfying immersion experience according to the device used. The VR player will have an illusion of being inside the city therefore their task is to navigate around the city and collect pumpkins from the minions. The mobile phone users can see the entire city and also the VR player to maximize the use of touch screen. As they would not get the feeling of being inside the city, they are assigned the task of laying traps for the VR player.

### 2.2 Hardware & Software

**Hardware:**
This application will be best experienced if the following devices are used:

1. HTC Vive – HTC Vive is one of the best VR headsets in the market and it comes with 2 controllers and 2 base stations. The sensors need to be placed diagonally from each other with a maximum distance of 5m. The room where the Vive is set up needs to at least measure 2m x 1.5m [2].

2. Android/iOS mobile devices – Any Android device with Lollipop 5.x and above and iOS device with iOS 10 would give the best results for playing the game.

3. Windows desktop PC or laptop [3] – HTC Vive is compatible only with Windows at this stage and therefore, for this project, only Windows PC would be well-suited. The absolute minimum requirement of Windows system are:
   - Graphics processor: Nvidia GeForce GTX970, or AMD Radeon R9 290 equivalent or greater.
   - CPU: Intel i5-4590 or AMD FX 8350 equivalent or greater
   - RAM: At least 4GB
Video output: HDMI 1.4 or DisplayPort 1.2 or newer
USB port: One USB 2.0 or greater
Operating system: Windows 7 SP1 or newer

Software:

1. Unity 3D engine – Unity is a powerful engine to develop games and get different executable file for various type of devices. Therefore, this project would be developed on Unity.
2. Unity Steam VR plugin – This plugin implements basic Vive related controller methods for new developers to use and modify. This will help this project’s initial set up.
3. Photon engine [4] – This will be used for networking between the different type of devices as it is very powerful and has a lot of features like voice chatting for future development improvements of this project.

2.3 Implementation

At this stage, for this project, progress has been made on the server architecture, research on the Vive controllers, camera positioning, and algorithm. This section describes in detail regarding the development in these areas:

Server Architecture:

In this game, there will be extensive networking involved in the virtual environment whenever any player takes an action. For examples, if the VR players’ avatar in virtual world moves and shoots any of the mobile phone player, then this should be updated on all the devices in real time.
Figure 1: Architecture diagram for networking between different types of devices.

The game server registers players basic information and adds him to the list of players. For the first player who joins the game, the server makes him the master client and all the other player who join the game connect through this client. As seen in the Figure 1, all the actions taken by the players are first sent to the master client, and then the game state is updated on the client. This updated information is sent to all the players connected to the master client. The game server will be responsible only for the creation of master client and a room. In this project, photon engine, a plugin for Unity is used to implement the game server. This engine is used because it can handle the server logic quite well for desktop and mobile multiplayer games. It generates a unique code for ‘n’ number of devices. These codes are sent to the ‘n’ devices who want to connect to the game. It handles the actions performed by these devices by their unique codes to differentiate among different players.
Vive Controllers

The front end of the game will be developed using C# in Unity. This is mainly because the Steam VR plugin is written in C#. This plugin will help to set up basic events for the game such as teleportation, using controllers to shoot etc. The number of things that the VR player can do is dependent on the Vive controller.

Figure 2: HTC Vive controller.

Figure 2 shows a table of the functions of the interactive options on the controller. For example, the part of the controller marked as 1, is a menu button. For this project, the menu button (1), tracking sensor (6), trigger (7) and grip button (8) will be used. As there are 2 controllers that accompany Vive, the left controller will be shown as a broomstick in the virtual world for the witch. The player can press the trigger button and point in a direction he/she wants to go in, and he/she will be transported to that location. When the player points in a direction, there will be a laser pointer originating from the broomstick which will follow projectile motion. The player will get transported to the point obtained by the intersection of the laser pointer and the surface the player is standing on. The right controller will serve a different purpose. The player can
use the right controller’s the trigger button to pick up objects like pumpkins or bombs and interact with them in the scene. The grip button will be used to shoot fire balls at the enemies. For both the controllers, the tracking sensor gives information about the position of the controller in the room to the game engine. This is done with the assistance of the base stations which are set up in the room to track the positions of the controller.

**Camera**

As stated in the section 2.1, the mobile user will view the game scene in a third person view and the VR player will get the first person view. This will be achieved by placing 2 cameras in the scene.

![Camera views for Mobile and VR player.](image)

**Figure 3: Camera views for Mobile and VR player.**

Figure 3 illustrates the camera position of the two types of players. Camera 1 will be placed at a fixed position in the scene which would render the game state for the mobile phone players. They can, however, zoom in the specific location of the scene to get a detailed view of that region. Whereas the VR players camera, camera 2, will be attached to its head position which will
render only the parts at a certain field of view. This helps the VR player get illusion of being physically present in the scene.

**Algorithm**

The algorithm for the game logic includes an Artificial Intelligence which will be based on path finding algorithms like A*, Depth First Search, Breadth First Search. These algorithms take the origin and destination into account and gives the shortest path from the origin to the destination. For this project, A* would be used because it gives more flexibility to the developer in implementing this algorithm as it considers cost of finding the shortest path and cost of every alternative path taken. There will always be 8 minions in the scene. Therefore, if only one mobile layer joins the game, remaining 7 minions will be controlled by the AI. If there is no mobile phone player, the AI will control all the minions in the city and try to defeat the VR player. However, the difficulty laid by the AI will depend on the number of minions controlled by it. This will guarantee fairness to the players and both the types will have equal chances of winning in the game.

**2.4 Testing**

The testing strategy is based on the feedback loop concept provided by XP. As soon as a result is achieved, XP suggests to get feedback from the clients, in this case potential players.

Therefore, to test the various developmental stages of the game, some potential players will be asked to play the game. Their feedback will help determine if the stage of development is complete or not. For the first stage of this project, the game story was pitched to many students who are aware about VR. Their feedback helped change a few details of the game and when they were satisfied with the changes, the game story and design was marked as completed. For the third stage, the users will be asked to test the ease of using the controllers and movements across the city. Some points on how they will hold the controllers and how much time it will take for them to use the controllers as required will be noted. Improvements to the pace of the game will be made accordingly.
For the final version of the game, in order to test it, at least 3-4 people are needed to judge the implementation of networking capabilities. Also, since this game requires usage of HTC Vive, and considering the fact that Vive is expensive, not many people would own it, there would be a demo set up outside a popular restaurant on The University of Hong Kong campus. They would be asked to try the game and detailed observation about their reactions to the different elements in the game would be recorded. They will be asked to fill out a survey to get their overall comment on the user experience of the game. The changes would be made to the game based on the received feedback.
3. SCHEDULE

3.1 Timeline

<table>
<thead>
<tr>
<th>Stage</th>
<th>Milestone</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game Design</td>
<td>Game story &amp; Mechanics</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Improvements</td>
<td>Completed</td>
</tr>
<tr>
<td>Networking</td>
<td>Testing Networking capabilities</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Improvements</td>
<td>Completed</td>
</tr>
<tr>
<td>Player Movements</td>
<td>Mobile movements</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Mobile data networking</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>VR movements</td>
<td>In Progress</td>
</tr>
<tr>
<td></td>
<td>VR data networking</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvements</td>
<td></td>
</tr>
<tr>
<td>UI</td>
<td>Scene Setup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Character animations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Game Panel (points, scores)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvements</td>
<td></td>
</tr>
<tr>
<td>AI</td>
<td>AI controlling minions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvements</td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td>Demo for the additional features</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvements</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Project timeline.
Table 1 describes a detailed timeline for the project development. The stages describe the mark point which ends the iteration of that phase. The milestones are the small deliverables for the supervisor to review.

3.2 Current Status

Refer to Table 1 for the stages described in this section.

1. By November, the game design stage was completed. The game story was designed so that the models from the master’s students’ work could be re-used. Some Computer Science (CS) students were asked to give feedback on the story and improvements were made accordingly.

2. Testing the feasibility of playing the game with multiple players to match the game story was completed by the end of December. It was tested with the same group of CS students.

3. Half of the third stage of the project is completed and third stage is expected to be completed by the end of January.
4. RISKS, CHALLENGES AND MITIGATION

<table>
<thead>
<tr>
<th>Risks/Challenges</th>
<th>Mitigation Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Challenge</strong> – sensitivity of the controllers tracking</td>
<td>Change the model of the controller to something players can connect with.</td>
</tr>
<tr>
<td><strong>Challenge</strong> – some studies claimed VR applications are not used by players</td>
<td>Implement game logic to be finished under 30 minutes or research on how to make</td>
</tr>
<tr>
<td>continuously for more than 30 minutes</td>
<td>the experience better and the players continue to play and enjoy the game.</td>
</tr>
<tr>
<td><strong>Risk</strong> – server side scripting interruption due to calls on mobile phone or slow</td>
<td>Write all the possible test cases and make sure the application can handle all of</td>
</tr>
<tr>
<td>internet connection.</td>
<td>them.</td>
</tr>
<tr>
<td><strong>Challenge</strong> – Online multiplayer compatibility requires chat or voice call</td>
<td>Future development due to time restrictions</td>
</tr>
<tr>
<td>option to communicate/co-ordinate among the players</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Risks, Challenges and Mitigation

5. CONCLUSION

This project delivers a multiplayer game application which can be played with different types of devices like mobile phones and Vive. This paper described the first and second stage of Extreme Programming game development cycle. The feedback loop ensures timely deliverable and aids in avoiding risks and challenges early in the development. The game design and networking feasibility has been tested and the later stages such as UI and AI algorithm for the game will be implemented.

Future work for this project would include online multiplayer compatibility and if the time allows for this version of the project, there would be a VR – Google Cardboard mode for mobile users in which they can choose to interact with Virtual world using touch screen or mobile phone sensors in the VR mode.
6. REFERENCES


