Hui Sen Fung Felix
3035090327

Supervisor:
Dr. T.W. Chim

Groupmates:
Wai Yip Yin Calvin 2008993564
Lau Chui Shan Sandy 3035097325
Leung Ho Ki Tony 3035097210
SUMMARY

Virtual reality technology has received growing attention in the entertainment industry in recent years. Yet there are still numerous problems in areas such as locomotion method that affects user experience in virtual reality games. Also, we believe that virtual reality technology can provide more useful applications other than solely gaming. Hence the project team proposes to develop an role playing virtual reality game that is specifically designed to provide an immersive wilderness training experience to the player, along with an affordable and light-weight locomotion solution.

This report investigates into the current state and problems of virtual reality games and locomotion control methods, and outlines the project design and development process in two separate threads: (1) training game development, (2) motion capture and locomotion method. It then illustrates future plans for the project.
ACKNOWLEDGMENT

I would like to express my sincere gratitude towards Dr. T.W. Chim for supervising our team for our Final Year Project. Dr. Chim had oversees our project throughout the year from project planning to the development stage. He helped us in realizing the vision of building the virtual reality training program and provided us insights on the motion capture problems. We could not finish this project without his guidance. I also thank Mr. Ronald Chung who was the second examiner of the project and gave us valuable feedbacks on the project.

I would also like to thank the developers of the MS-Kinect SDK for Unity that is an essential integral component in our project.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Background</td>
<td>7</td>
</tr>
<tr>
<td>2. Purposes</td>
<td>8</td>
</tr>
<tr>
<td>3. Objectives</td>
<td>9</td>
</tr>
<tr>
<td>4. Previous Work</td>
<td>10</td>
</tr>
<tr>
<td>A. Virtual Reality Training</td>
<td>10</td>
</tr>
<tr>
<td>B. Locomotion Methods &amp; Input Control</td>
<td>11</td>
</tr>
<tr>
<td>5. Project Overview</td>
<td>16</td>
</tr>
<tr>
<td>A. Game Development</td>
<td>16</td>
</tr>
<tr>
<td>B. Locomotion Methods &amp; Input Control</td>
<td>16</td>
</tr>
<tr>
<td>Contributions</td>
<td>16</td>
</tr>
<tr>
<td>6. Methodology</td>
<td>17</td>
</tr>
<tr>
<td>A. Overall Game System</td>
<td>17</td>
</tr>
<tr>
<td>B. Game Development</td>
<td>18</td>
</tr>
<tr>
<td>C. Locomotion Methods &amp; Input Control</td>
<td>20</td>
</tr>
<tr>
<td>7. Experiments</td>
<td>22</td>
</tr>
<tr>
<td>8. Implementation</td>
<td>23</td>
</tr>
<tr>
<td>A. Game Development</td>
<td>23</td>
</tr>
<tr>
<td>B. Locomotion Methods &amp; Input Control</td>
<td>27</td>
</tr>
<tr>
<td>9. Conclusion</td>
<td>36</td>
</tr>
<tr>
<td>10. Future Work</td>
<td>36</td>
</tr>
<tr>
<td>A. Game Development</td>
<td>36</td>
</tr>
<tr>
<td>B. Locomotion Methods &amp; Input Control</td>
<td>37</td>
</tr>
<tr>
<td>References</td>
<td>38</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Caption</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4.1</td>
<td>VR medical training developed by RCSI</td>
<td>10</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>British military VR training</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>A woman using Oculus Rift gadgets</td>
<td>12</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>HTC Vive setup</td>
<td>14</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>A Virtuix Omni platform</td>
<td>15</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>Overall game system structure</td>
<td>17</td>
</tr>
<tr>
<td>Figure 6.2</td>
<td>Kinect sensor structure</td>
<td>21</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>Joints position when walking</td>
<td>22</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>Pick up object posture a</td>
<td>23</td>
</tr>
<tr>
<td>Figure 7.3</td>
<td>Pick up object posture b</td>
<td>23</td>
</tr>
<tr>
<td>Figure 8.1</td>
<td>Shelter building steps</td>
<td>24</td>
</tr>
<tr>
<td>Figure 8.2</td>
<td>Inventory menu</td>
<td>25</td>
</tr>
<tr>
<td>Figure 8.3</td>
<td>Player avatar model and skeleton</td>
<td>26</td>
</tr>
<tr>
<td>Figure 8.4</td>
<td>Game environment overview</td>
<td>27</td>
</tr>
<tr>
<td>Figure 8.5</td>
<td>Function code for handling player-turning around situations</td>
<td>28</td>
</tr>
<tr>
<td>Figure 8.6</td>
<td>Kinect sensor axis</td>
<td>30</td>
</tr>
<tr>
<td>Figure 8.7</td>
<td>Pick up posture detection code</td>
<td>30</td>
</tr>
<tr>
<td>Figure 8.8</td>
<td>Walking gesture state transition diagram</td>
<td>31</td>
</tr>
<tr>
<td>Figure 8.9</td>
<td>Walking gesture detection code</td>
<td>32</td>
</tr>
<tr>
<td>Figure 8.10</td>
<td>Body trunk area</td>
<td>33</td>
</tr>
<tr>
<td>Figure 8.11</td>
<td>Menu manipulation state transition diagram</td>
<td>34</td>
</tr>
<tr>
<td>Figure 8.12</td>
<td>Motion platform drawing</td>
<td>35</td>
</tr>
</tbody>
</table>

LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 5.1</td>
<td>Teammates Contributions</td>
<td>17</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>List of tutorial tasks</td>
<td>19</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Meaning</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>VR</td>
<td>Virtual Reality</td>
<td></td>
</tr>
<tr>
<td>RPG</td>
<td>Role Playing Game</td>
<td></td>
</tr>
<tr>
<td>OWRPG</td>
<td>Open World Role Playing Game</td>
<td></td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
<td></td>
</tr>
<tr>
<td>MoCap</td>
<td>Motion Capture</td>
<td></td>
</tr>
<tr>
<td>MS</td>
<td>Microsoft</td>
<td></td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
<td></td>
</tr>
</tbody>
</table>
1. BACKGROUND

Virtual reality is the technology that uses computer generated sensations to replicate a realistic environment for the user. In recent years, virtual reality has been made more realistic and more easily accessible. Player interaction functions were also largely enhanced with the use of specialized hardwares such as controllers and motion sensors. However, there are still some problems found in existing virtual reality games: unnatural means of locomotion controls and pricely gadgets.

In order to create a highly immersiveness experience, the two major elements, perception and interaction, must be well implemented. Nowadays, perception in virtual reality software is largely enhanced by high quality graphics. But the interaction between the player and the game world remains unnatural. One of the reason is that the locomotion methods that most virtual reality software uses are unnatural when compared with how people really move in the real world. For example, some games only allow users to use head movements as game input. This does not only limit the game interaction options, but it also severely damage the immersiveness of the game experience.

As the virtual reality technology is enhanced every year, most applications developed with VR technology are for entertainment purposes, i.e. gaming and videos. There are indeed VR applications that facilitates military and medical trainings. But they are not accessible or useful to the general public. Virtual reality technology allows people to explore environments and situations that could not be easily or safely accessible. Using virtual reality only for entertainment purposes would be a waste of its advantages and potentials.
2. PURPOSES

There are currently two main problems of the use of virtual reality technology. The application problem is the lack of use of VR technology in fields other than entertainment, while the technical problem is the unnatural locomotion method.

While most virtual reality software is developed in the entertainment field, there is great potential for virtual reality to be used for educational and training purposes. Users can scrutinize dangerous areas or uncommon incidents in first-person perspectives without having to encounter the danger and discomfort that could happen in real life situations. For instance, there exists some military and medical VR trainings because wars scenarios are extremely dangerous and it would cause serious harm to patients if delicate medical situations are handled poorly. These situations that can easily go wrong can be experienced by virtual reality technology. And because VR environment is realistic, it can provide a safe training that feels much more similar to real practice than other alternative training methods.

The use of unnatural locomotion methods will not only damage the immersiveness of sensations, but also can cause motion sickness to some users. The locomotion methods used in currently existing virtual reality software can be generally divided into two categories: artificial locomotion and real-world movement. Artificial locomotion makes use of controllers for in-game navigation, for example button controls, teleportation or mimicking of arm swinging movements when walking. Despite it is easier to implement these methods, they do not provide an immersive game experience. It can easily cause serious motion sickness due to the inconsistency between the images we see and the actual movement our body. So to address these
issues, some VR companies have developed real-world movement locomotion solutions with motion capture systems. It allows players to actually walk and move in a natural way on a static virtual reality platform and converts their body gestures into control inputs. However, these motion platforms and motion capture systems are often quite expensive and large in size. Hence even it provides more immersive user experience, it is not a desirable solution as it hugely affects its dissemination.

The aim of this project is to develop an immersive virtual reality role playing game that provides educational purposes with the use of an affordable but realistic real-world movement locomotion method. We chose wilderness survival as our theme for this project. Most people nowadays live in the city where they lost all their basic survival skills. Hence we hope to provide an outdoor survival training to our users. VR technology fits perfectly with this theme because outdoor survival could be dangerous and VR can create a safe environment for users to gain hands-on wilderness survival experience. From this, we could investigate the likelihood of using virtual reality technology for general education purposes. In addition, we would also develop a locomotion method that utilize our body movement as game inputs and it should be cheaper than most of the current popular locomotion methods.

3. OBJECTIVES

The objectives for this project are twofold:

- To design and develop a virtual reality game dedicated for providing wilderness training
- To implement an affordable whole-body motion capture system for game interaction and locomotion purposes.
4. PREVIOUS WORK

A. Virtual Reality Training

i. Healthcare

(Fig. 4.1) VR medical training developed by RCSI.

Medical practices especially surgery trainings are often difficult to be held. Because medical students are often unexperienced and unqualified for performing medical procedures on patients. But how can they learn without any trainings? In most medical schools, students learn from videos and plastic models. Virtual reality technology can improve the learning process by letting user stand in the shoes of a medical staff. RCSI had developed a VR medical training simulator for medical trainees to experience the assessment and treatment procedures done to a patient. User can thus learn to make medical decisions without causing any real-life troubles.
The VR application is a good way for surgery trainees to learn through experience in the stimulated and safe environment. But the application is lacking interactive element between user and the application.

ii. Military

(Fig. 4.2) British military VR training.

War zones are extremely dangerous and soldiers could perform very differently with how they did in trainings. So practicing in a safe environment may not help them prepare for real situations. UK military uses VR technology for building a virtual war zone so that soldier will train in a seemingly dangerous environment and undergo the pressure similar to real-life experience. This training software is far more advanced compare to general VR war games. A group of soldiers will collaborate with each other in a large room with many sensors. It is successful because it is able to create an immersive stimulation that soldiers believe that they are in a real war zone. They will have the similar psychological reaction to that when they are in the real situation.

B. Locomotion Methods & Input Control
Oculus and HTC are the industry-leading companies in virtual reality gaming. They each have a different approach in developing their locomotion methods. Oculus uses various types of artificial locomotion methods, while HTC focuses on real-world movement for locomotion methods. They developed Oculus Rift and HTC Vive respectively with their chosen locomotion approaches.

i. Oculus Rift

(Fig. 4.3) A woman using Oculus Rift gadgets.

Oculus Rift is designed for people playing VR game sitting down in front of the computer. Hence the tracking system is optimized for standing or sitting experience. The Oculus Rift package includes a VR headset, a game pad, an Xbox controller and one table-standing sensor. As Oculus wants their users to sit comfortably on a chair when they are enjoying their VR experience, the setup heavily relies on the buttons on the game pad or Xbox controller as game input. All the game interactions are abstracted and can be easily accessed via buttons, or via the tracking of the one game pad.
Oculus later introduces Oculus Touch controllers that allow tracking of hand movements. So some developers have created other artificial locomotion methods for in-game navigations with only hand movements. For example, the teleportation method helps player to be transported to a location pointed by the controller. There are also methods that convert arm swinging movements into walking movements in the game world while the moving direction will be determined by the controllers. But this method will sometimes confuse certain arm movements with the arm swinging motion during walking.

The Oculus Rift offers easy and convenient gameplay and control setup. It is capable of letting people play in a small area. But there are several drawbacks in its approach. Firstly, there are many buttons on the Xbox controller and the Touch controller. Players need to remember where each button is and what the button does. These artificial locomotion methods would make them more aware that they are not in the real world and cause immersive issue. It is more likely for players to feel nausea during gameplay. Also, the price is not cheap. The hardware package costs USD $600, and the Oculus Touch controllers cost an extra USD $200.

ii. HTC Vive
HTC believes the best option for virtual reality is real-world movement. Hence their gadgets allow people to walk around in a roomscale-tracked space. HTC Vive includes VR headset, two hand controller, and two sensors called Lighthouse beacons. The sensors will be placed at two corners of the room, tracking player movements in a 16x16 feet area. Player can thus walk freely in the area, enjoying a immersive interaction with the virtual world. This method also largely reduces the discomfort brought by motion sickness.

Despite the experience and freedom provided by HTC Vive seem to be better than Oculus Rift, the player movement is still limited in 16x16 feet area. This means the player would not be able to navigate far in the virtual world if they do not use artificial locomotion method as remedy. Although it is perfect for some sports games like tennis, it would cause a lot of troubles in an open world game. Setting up the tracking area is also hugely inconvenient because 16x16 feet of free space is hardly available in most people’s home, especially for Hong Kong people. Furthermore, the HTC vive package costs USD $800, which is very expensive.
Virtuix Omni differs from the above two packages in the way that beside using sensors and controllers, it includes a treadmill which simulate the motion of walking. The sensors will then record body movements. The treadmill does not only let user to freely walk on it indefinitely, its robust support ring can hold the player and ensure his safety. Hence, its ability to provide the most freedom gives users the most immersive experience among the three options. Yet the best solution comes with the most expensive price. The Virtuix Omni package costs USD $1000, while not including VR headsets and controllers. The platform is also large in size and difficult for storage.
5. PROJECT OVERVIEW

A. Game Development

We planned to develop a wilderness survival game named “Powerless” for this project. The aim of this game is to train and teach users the necessary skills and knowledge in surviving in the wild. The game will be developed for virtual reality experience, and users should be able to interact with the game world and navigate freely in an open world. Users body movement will be used as game inputs and image will be presented through the VR headset.

B. Locomotion Method & Input Control

The locomotion method developed should be able to translate movements of player’s lower limbs as navigation input. Because body movements can reduce motion sickness and enhance immersiveness. The setup should be able to be used in a room at home and should be affordable. The movements allowed as control input should be as natural as how people move in the real world. A motion capture system would be used to detect movements, and our game implementation will turn the movement data into gesture and relate them to corresponding functions. A simple, easy-to-install motion platform would also be constructed to ensure player’s safety and prevent them from walking out of the motion detecting area or hitting a wall.

Contributions

Our team consist of four members and each of us is responsible for the following parts:
<table>
<thead>
<tr>
<th>Members</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hui Sen Fung Felix</td>
<td>Gesture Recognition Module Development</td>
</tr>
<tr>
<td></td>
<td>Input Control Design</td>
</tr>
<tr>
<td></td>
<td>Locomotion Method Implementation</td>
</tr>
<tr>
<td></td>
<td>UI Development</td>
</tr>
<tr>
<td>Leung Ho Ki Tony</td>
<td>Game Design</td>
</tr>
<tr>
<td></td>
<td>Skill Research</td>
</tr>
<tr>
<td></td>
<td>Platform Development</td>
</tr>
<tr>
<td>Lau Chui Shan Sandy</td>
<td>3D Modelling &amp; Animation Production</td>
</tr>
<tr>
<td></td>
<td>Stage Visualization</td>
</tr>
<tr>
<td>Wai Yip Yin Calvin</td>
<td>Game Logic Programming</td>
</tr>
<tr>
<td></td>
<td>Code Review</td>
</tr>
<tr>
<td></td>
<td>Platform Development</td>
</tr>
</tbody>
</table>

(Table 5.1)

6. METHODOLOGY

A. Overall Game System

(Fig. 6.1) Overall game system structure.

The game requires a number of components to work. The game will be executed on a computer. The graphics of the game will be streamed to a mobile phone via a VR streaming application called KinoVR using Bluetooth or Wifi, and the player will be able to enjoy the VR experience with a VR headset. This setup allows the game to run
smoothly with the computer’s computational power and connect to the sensor. The sensor will capture player’s body motion as control input and the game will differentiate the gestures performed and translate into certain functions. The player can use the platform to ensure their safety when they are playing the game.

B. Game Development

The game development process was mainly divided into 4 phases: design, implementation and testing.

i. Design

The first step of the design phase was to do research on outdoor survival skills so that we could learn what needs to be included in the project, such as the gestures performed and the required game objects. Based on the findings and our discussion, the four elements of game design are specified as follow:

Technology: Game development would be done by Unity and modelling by Blender and MakeHuman; Kinect is used body movement tracking device; KinoVR is used to stream game graphics to mobile phone in the headset.

Story: The character is left alone on an isolated island, with no modern tools. But there are abundant resources and animals on the island. The player will need to provide for himself with handmade tools. He will have to do tasks to survive, which includes building shelter and making fire. Each of these tasks consists of several subtasks. The detail of each tasks are listed as below:

<table>
<thead>
<tr>
<th>Tutorial tasks</th>
<th>Subtasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources Collection</td>
<td>▪ Pick up nearby objects</td>
</tr>
<tr>
<td></td>
<td>▪ Drop things</td>
</tr>
<tr>
<td></td>
<td>▪ Collect 10 tree branches</td>
</tr>
<tr>
<td>Building Shelter</td>
<td>▪ Collect debris</td>
</tr>
<tr>
<td></td>
<td>▪ Find a ridge pole</td>
</tr>
<tr>
<td></td>
<td>▪ Find 2 “Y”-shaped sticks</td>
</tr>
</tbody>
</table>
- Prop up the ridge pole
- Build the framework

### Starting Fire
- Collect tinder
- Gather kindling
- Make nest with tinder
- Make teepee with tree branches on the nest
- Get a fireboard
- Make a drill
- Make a bow
- Attach tree branch on bow string and drill on the fireboard
- Put fireboard to tinder nest when smoke appears

### Hunting
- Collect a long thick tree branch
- Sharpen it into a spear
- Hide
- Wait for animals
- Kill with the spear
- Stop the bleeding
- Leave and clean
- Return to shelter

### First-aid treatment
- Raise wounded area above heart level
- Stop bleeding
- Hold
- Clean the wound
- Wrap the wound

(Table 6.1)

**Mechanics:** The aim of the game is to complete all the survival tasks. The player can do whatever he wishes in the game world, but instructions will be provided to guide the players through the tutorial. The player has to avoid injuries, hunger and thirst. These will contribute to the health level. Once the health level drops to zero, the character will die and the game will be over.

**Aesthetics:** The player will play the game in first person position with virtual reality technology. Modelling and animations will be done by Blender and MakeHuman.

**ii. Implementation**
The implementation work would be divided into three parts for easier job allocation: game environment, game mechanics, locomotion and gesture input. Game environment and objects would be built with scripts describing their attributes. Game
mechanics would focus on writing scripts on player behavior and item characteristics, and implementing game master manager that controls overall game logic. Locomotion method and input control will be discussed in a later session.

The team follows agile methodology on game development. The project was developed and reviewed iteratively. The first sprint focused on player attributes (i.e. health scripts). The following sprints implement each stage tasks.

iii. Testing
Every function is tested by testing scripts before moving on to the next task. This can reduce debugging time as the project grows bigger in scale. The three separate parts are also tested before putting together. Test plays will be conducted on the locomotion method and gesture recognition before integration. Another round of test plays will be held to receive feedbacks on the overall game for modification.

C. Locomotion Method & Input Control
The first step in developing our locomotion method is to select a suitable motion capture system. Initially inertial measurement unit (IMU) for MoCap was selected accurate 360° body sensing. However, IMU systems are expensive (Noitom’s perceptive neurons costs USD $800). Hence Kinect was selected as an alternative motion sensing devices.
Microsoft Kinect is made for body sensing in Xbox games. It can identify human body and extract a number of skeletal points on human body from the image it perceives. It also has a depth camera so it perceives objects as 3D instead of 2D. One of the major reasons Kinect is chosen as our motion sensor is because it fits our affordability requirement. It costs only USD $100 and it is available in the CS department in many universities such as HKU. In addition, it does not require a lot of space (2m distance from Kinect). But to use Kinect for locomotion method and control input, the 2 drawbacks of Kinect need to be addressed. (1) It is designed for front facing games only. So in an open world game, a person cannot decide his navigation direction. (2) Without hand controller, it is unable to call for certain in-game functions, such as opening a menu.

To solve (1), gyroscope of the phone in the VR headset is used for determine the direction the player is facing. It was used with Kinect for locomotion method. For (2), gesture recognition would be used to call certain game functions. A wireless mouse would also be held to give more varieties of input control.
7. EXPERIMENTS

Posture & Gesture Study

To better understand the walking movements for gesture recognition, data of frequently performed actions are recorded. This can also facilitate platform design.

i. Walking

(Fig. 7.1) [a] Joints position when walking in place; [b-c] Walking forward.

Figure 7.1(a) shows a person walking in place, while (b) and (c) demonstrate how the postures involved when the person walks naturally. The walking movement consists of two gestures: pulling up leg and put down foot. It was discovered that the position of a foot is not necessarily much higher than the other one. When a person is walking in place, the distance between two feet is also quite small. Arm swinging movements can also be easily confused with other hand gestures. So instead, the change in distance between knee joints is more significant when both walking naturally and walking in place. Hence it is more desirable for detecting walking movements.
ii. Pick up items from ground

(Fig. 7.2) Pick up object posture a. (Fig. 7.3) Pick up object posture b.

Figures 7.2 and 7.3 present how people pick up objects from the ground. Some people bend forward to pick things up, while some crouch before they reach for the object. This means the gesture recognition implementation has to consider both cases. And since the action might involve leg movements (person in Figure 7.2 bends his knees slightly sideways), implementation should be careful not to confuse it with walking movements.

8. IMPLEMENTATION

A. Game Development

i. Game Mechanics

1. Item behaviors

There are more than 10 usable objects in the game. Each of them possess different roles in different tasks. There are several types of scripts to define their roles and attributes.
Object handler: describes the type and parameters of the object.

Pickables: a class created to identify the object that can be picked up and used by the player. These objects will be able to be placed in the inventory.

Building handler: the script was written for shelter building task. It is attached to certain building materials and handles the triggering condition that build the structure component. The conditions include collision and removal of materials. When the condition is satisfied, the resulting structure will appear and initial materials will be destroyed.

(Fig. 8.1) Shelter building steps.

Skill handler: the skill handler was created for the fire starting skill. When the player is holding the drill and the bow, the skill handler will call a gesture detection function that listens to the player’s fire drilling movement. If the movement is continuous and fast, the temperature will continue to raise until smoke comes out.

2. Player behaviors
The player behavior scripts deal with the interactions between the player and other game objects, and also record the attributes of the character.
Player statistics handler: the handler records and manages the hunger and thirst level, and overall health points of the player. Hunger and thirst level will drop bit by bit and player has to find food and water to raise the levels. Otherwise the player will have a negative body condition which affects its movement or even lead to death.

User Interface manager: the player’s health condition needs to be acknowledged by the player so that he will know when to improve his health conditions. Player statistics handler will be collected and displayed on the screen. The tutorial instructions is also saved here and called at the suitable time.

Inventory manager: the inventory menu acts as a virtual backpack that opens when the player reaches for his backpack. It records all the items the player possessed. The player can select objects and choose to use or drop the object. These functions will be called by gesture recognition script. When a player picks up an object, the item will be directly saved to the inventory.

(Fig. 8.2) Inventory menu.

Avatar controller and gesture recognition (to be discussed in the next session)

3. Overall game logic
The overall game flow is managed by the game master and game controller. The
game master is the highest level game manager that controls the game mode.
There are currently two game modes: quest mode and survival mode. The quest
mode is managed by the game controller. It will guide the player to complete the
tutorial tasks one by one.

ii. Game object modelling

1. Player avatar

(Fig. 8.3) Player avatar model and skeleton.

MakeHuman was used in the production of player avatar 3D model. The avatar
produced contains all the skeletal structures and can be easily mapped to Kinect
data. During Kinect testing, it is found that the head would obstruct the first
person camera. Hence the head was removed from the model part while the
skeletal structure was kept intact.

2. Environment
Based on the stage designs, there are a few elements required for the environment, such as forests, river and rocks, as tree logs, water and stone are needed for making tools or shelters for survival. Cliffs, large rocks and the sea surround the game environment to discourage people from leaving the area.

The island is built with terrain object in Unity3D, while some objects in the environment were imported from the online asset store. The tools or structures that are built from multiple components are not remodeled in Blender. Instead, they are combined together as a prefab. This reduces development time and game model storage space.

**B. Locomotion Method & Input Control**

i. **Navigation Direction**

Since Kinect cannot determine accurately what direction the user is facing, especially front and back, a gyroscope from the phone in the headset was used to help determine the navigation direction.
(Fig. 8.5) Function code for handling player-turning-around situations, currently commented out. `turnAroundHandler()` was written so that when the user turns left or right for more than 90° from its initial direction, the avatar would display mirrored movements of the Kinect-perceived skeleton, because Kinect would be looking at the back of the user. This method succeed in matching the correct joints to each body parts when the user turned around for 180°. However, the avatar was messed up when the user turned its body 90° from the Kinect’s viewing direction. Because body parts overlap when the body is facing sideways. Kinect can only see one leg and cannot differentiate which leg it is. Kinect is also unable locate the arms if they are put on the side of the body as it can only see one big trunk. Hence Kinect will try to find the hidden body parts by trying to include background objects into the skeleton, causing the avatar to mess up.

At this point there were two ways to continue the locomotion method development. The first solution was to put the Kinect on to the platform, and make it able to rotate when the player turns, so that the player would always face the Kinect sensor, while the gyroscope would determine the navigation direction. This method would allow the player turn 360° without any Kinect sensation problems or player movement restrictions. But this means the locomotion method heavily relies on the platform. The other approach to the problem is to restrict player’s movement. Turning body to sideways would not be allowed. Instead the navigation direction would be determined solely by head movements, as it controls the gyroscope. To navigate

```csharp
327     Vector3 setInitialDirection(){
328         return Camera.main.transform.forward;
329     }
330
331    void turnAroundHandler(Vector3 initDirection, Vector3 camDirection){
332        initDirection.y = 0;
333        camDirection.y = 0;
334        float turningAngle = Vector3.Angle(initDirection, camDirection);
335        GameObject player = GameObject.FindGameObjectWithTag("Player");
336        AvatarController avatarController = player.GetComponent<AvatarController> ();
337        if (turningAngle > 90) {
338            print("mirrored");
339            avatarController.mirroredMovement = true;
340        } else {
341            avatarController.mirroredMovement = false;
342        }
343    }
344```
backwards, a mouse button would be clicked to turn the avatar 180° backwards so the user could navigate at all directions. This method would slightly affect the immersiveness, but it guarantees the safety of the player and requires little assistance from the platform.

The second control method was selected because the first method requires heavy development of the platform and the team lacks the skills for that. The Kinect might not be able to sensor properly in a very close range as well. In addition, high complexity of the platform means the cost of platform increases. On the other hand, “click to turn” method fits our requirement on the cost and safety. The player is still allowed to move their body despite certain restrictions. This also enables player to play without a platform. But still, our team had worked on a platform according to the first solution as it can provide the best immersiveness (discussed in the session “Motion platform”).

The navigation direction could now be determined. The detection of player’s walking movement would be discussed in the session “Gesture Recognition & Input Control”.

ii. Gesture Recognition Algorithms and Input Control
The data read from the Kinect will be used for input control and gesture recognition purposes. The avatar controller maps skeletal data received to the corresponding joints of the avatar, thus controlling the avatar’s movement. At this point the player can interact with the game scene with collider. To call more abstract functions, gesture recognition is used. As the features of walking movement and picking up action were studied in the posture and gesture study, gesture recognition could be implemented based on the information collected.
detectGesture() function is called once per frame. A Unity package “Kinect with MS-SDK” is used to read skeletal data from Kinect as Vector3 objects. Gestures are detected by comparing the relative positions of the joints with these vectors. Since a gesture often consists of a number of postures, a gesture detection is often divided into two or more parts, each detects and records a posture. If the postures are done consecutively, the gesture is completed and it will call the corresponding function.

1. Pick Up Gesture

```csharp
122  void detectPickUpPosture(Vector3 rHandPos, Vector3 lHandPos)
123  {
124      if (rHandPos != emptyVector & lHandPos != emptyVector) {
125          if (rHandPos.y <= 0.55 || lHandPos.y <= 0.55) {
126              if (Input.GetMouseButtonDown (0)) {
127                  print ("pick up");
128                  //TO-DO: pick up nearby items
129              }
130          }
131      }
```

(Fig. 8.7) Pick up posture detection code.

Picking up things from the floor is a frequently used function in a wilderness training as one needs to collect a lot of resources for survival. For gesture recognition, player is not required to really touch the ground to perform picking up action because it would be tiring for player to bend down too much and Kinect sometimes gives inaccurate coordinates and messed-up avatar when the person crouches. Initially we compare the y coordinates of the knee joints and the hand positions. After testing it is found that hand position can sometimes be lower than
knee joints when walking. So the final implementation compares the y coordinate of the hand positions to a threshold retrieved after repeated testing. And since some other activities such as making a fire requires crouching position and would trigger the pickup function, a mouse button click is included to prevent unwanted calling of the function.

Further investigation into how to avoid triggering walking gesture will be discussed in the next session.

2. Walking Gesture Detection Algorithm
The walking gesture consists of 4 postures: pulling up left leg, putting down left leg, pulling up right leg and putting down right leg. The alternate movements of the left leg and the right leg will form the walking gesture, and satisfy the conditions for walking detection. But since some players may move their legs a bit for other purposes (e.g. kicking, crouching), the player is allowed to move only after performing two walking step gestures to prevent some unwanted triggering of walking function.

(Fig. 8.8) Walking gesture state transition diagram.
The default state is “none”. If a leg is detected to raise over certain threshold, the state will change to “Up” with the side indicated. When the foot returns to the ground, the state will change to “Down”. At this point one step is performed. When the second step is performed by another leg, the move forward function will be called to move the player.

As mentioned in the experiment, the walking gesture is best measured by the knee joints. But as the distances between knees when standing and walking varies between individuals, it is difficult to set a threshold for comparison. Also, Kinect is a bit inaccurate in sensing the z position (depth) of perceived image. Thus the detection function takes in 6 joints (hip joints, knees joints and feet) for more accurate assessment. The perceived height (y) of the pulled up thigh is measured and the other thigh would be set as a threshold. The position of the feet is also measured to avoid confusion with crouch posture. To avoid unwanted triggers of the function, we set the required gesture a bit exaggerated from our natural walking actions.

3. Reach for backpack Gesture

(Fig. 8.9) Walking gesture detection code.
The reach for backpack gesture opens the item menu. To perform the gesture, the player would need to put one hand behind the back of the body, while the body trunk will be indicated by the left and right shoulder joints, and left and right hip joints as Figure 8.10. The drawback was that when a person is walking, his body may rotate a little on the x-z plane. Thus the hand might reach the detection area and open the item menu. To tackle this problem, a mouse click is required to trigger this function after performing this posture.

(Fig. 8.10) Body trunk area.

4. Menu Manipulation Gestures

Item menu manipulation often requires several control inputs, such as choosing previous or next item, use, discard and close menu. As these manipulation is not an actual action, gestures would be assigned to call each manipulation function. Swiping right and left would be sensible gestures to be assigned to choosing previous or next item. Player would swipe up to use and swipe down to discard. These arm gestures have to go from one side of the body to the other side. This means there are two postures in each gesture, one indicating the side the hand raised up from, one determining if the hand has swiped across the body.
Since player might not complete a full gesture and get stuck, a cancel function was designed to return to the “none” state. To cancel, player only need to put down both of his hands. To close the menu, the player need to put his hands together in front of his body trunk.

5. Pause Gesture
As there is no gesture that can relate to pausing, a gesture was assigned to call the pause function. The requirement for this function is that it would not be easily triggered by actions that people normally do. The designed gesture is to put both hands behind the neck as it would not be confused with any other gestures or activities.

iii. Motion Platform
In the posture and gesture study, the players’ waist heights were measured to help decide the height of the supporting pole. The supporting pole is linked to the base circular track and will hold the player on the platform with a belt tied to the waist. An opposite pole is also attached to the track to hold the Kinect sensor. When the player turns, the circular track would rotate and the Kinect pole will stay facing the player, so that the Kinect can always use its optimal sensing power.
9. CONCLUSION

The virtual reality gaming industry has been improving drastically in the past few years. Yet little effort had been put into educational field. Our team had built a tutorial game on wilderness survival with VR technology. It is successful in giving players hand-on experience on the survival skills and it is believed to be an effective way of learning high-risk skills. However, VR display does not allow large amount of instructions and information to be displayed on screen. VR can be a good training method, but other supporting learning materials are required.

While the current locomotion methods developed by the VR companies are either too abstract or too expensive, the team had built a locomotion method and gesture input control system for our own use with Kinect. It might not have a very high level of immersiveness as the gestures required are a bit exaggerated and the “click to turn” method is not ideal. But the solution does allow player to navigate in the open world conveniently with body movements and the solution built only requires a Kinect that costs only USD $100, which is one sixth of Oculus Rift’s price and one eighth of HTC Vive.

10. FUTURE WORK

A. Game Development

There are still a lot of survival skills that are not included in this project. Further developments can focus on other skill sets and others methods to achieve certain goal,
such as other ways to start a fire. New scenes can also be created for survival training in other harsh environments like desert and polar areas.

B. Locomotion Method & Input Control

After finding out the weakness of Kinect’s sideway detection, we had come to 2 solutions. Both of the solutions are far from a perfect solution. One overly relies on the function of platform while one is not entirely immersive as player’s movement is restricted. Further improvement on the locomotion requires more powerful motion sensors that is not limited to front facing sensing. Perhaps using two Kinect sensors that are set 90° apart around the player can improve the weakness in sideway body tracking. IMU is still the best option for accurate result and allows the most freedom of movements. However, these two solutions both cost at least twice as much as our current solutions. The gesture recognition module in the current project can also be used after modification and setting a local coordinate for the player as the player will be free to turn.
REFERENCES

- Swider M. [Internet]. HTC Vive vs Oculus Rift: which VR headset is better?; [updated 2017 Feb 6]; Available from: http://www.techradar.com/news/wearables/htc-vive-vs-oculus-rift-1301375
- RCSI [Internet]. RCSI VR Medical Training Sim; [cited 2017 April 6]; Available from: http://msurgery.ie/vr
- Parkin S. [Internet]. How VR is training the perfect soldier; [updated 2015 Dec 31]; Available from: https://www.wearable.com/vr/how-vr-is-training-the-perfect-soldier-1757