The University of Hong Kong
Faculty of Engineering
Department of Computer Science

COMP4801
Final Year Project
2016-2017

Group: FYP16025
POWERLESS
Open World Virtual Reality Role Playing Game

Individual Report

Prepared By:
Wai Yip Yin, Calvin (2008993564)

Supervisor:
Dr. T.W. Chim

Group Members:
Hui Sen Fung Felix (3035090327)
Leung Ho Ki (3035097210)
Lau Chui Shan (3035097325)
SUMMARY

Due to the production of virtual reality headset, virtual reality (VR) gaming has been receiving increased attention recently. Since then, console, mobile phone, and gaming gear corporations started entering the VR game industry by introducing their own VR gaming systems, including sophisticated controllers, sensors, and even platforms.

The gaming industry follows by publishing increasing amount of VR games for those systems, yet the results are not satisfying, as the immersion element of those games are greatly limited to the motion detection as well as game design.

As Hong Kong citizens, it is observable that newer generations are generally gaining fewer daily skills and knowledge than the past. While not hindering their life as a modern citizen, preserving skills and let future generations gaining those skills even if there is no teacher available would be beneficial if there is ever any needs to those lost abilities. It is suggested VR games could be a solution for archiving skills, letting players re-living the scenario which those skills are necessary or taught.

The project aimed to produce a game that is designed towards VR gaming, explore the potential benefits of VR games, and empower the player with the ability to explore the digital world.

ACKNOWLEDGEMENT

I would like to hereby express my greatest gratitude to Dr. T.W. Chim for supervising the project. With his kind guidance and attention, the development of the project was smooth yet still challenging. It is my pleasure to cooperate with Dr. Chim.

I would like to also express my sincere gratitude to Mr. Patrick Desloge for his kind teaching and comments on the proposal and interim report of the Team. The Team is able to improve greatly on delivering information and recording progress after his lecture.

I would like to deliver my gratitude to all the officials involved in the final year project course, who kindly provides supports to the Team.
TABLE OF CONTENT

1. Introduction .................................................. 3
2. Project Background and Literature Review.............. 4
   2.1. Background .............................................. 4
   2.2. Previous Work ....................................... 5
3. Project Scope .................................................. 10
   3.1. Game .................................................. 10
   3.2. Gesture Recognition .................................. 10
   3.3. Platform ................................................ 10
4. Project Methodology ............................................ 11
   4.1. Game Development .................................... 11
   4.2. Gesture Recognition .................................. 11
   4.3. Platform ................................................ 12
5. Implementation ............................................... 12
   5.1. Game Design .......................................... 12
   5.2. Project Hierarchy ..................................... 13
   5.3. Scripts ................................................ 15
   5.4. Visual Design ........................................ 19
   5.5. Gesture Recognition .................................. 22
   5.6. Platform ................................................ 23
6. Conclusion and Future Works ............................... 25
   6.1. Conclusion ............................................. 25
   6.2. Future Works .......................................... 25
Appendix: Reference List ........................................ 26

DEFINITION
AR shall mean Augmented Reality
HUD shall mean Heads Up Display
MVC shall mean Model-View-Controller
RPG shall mean Role Playing Game
UI shall mean User Interface
VR shall mean Virtual Reality
the Team shall mean the project group FYP16025
SECTION I - INTRODUCTION

With the development of VR headsets which incorporates mobile phone as the display and control device, VR gaming has been in a rapid development in recent years.

Current VR game installments, as limited by physical gaming space and equipment available to the general public, often require players perform actions not similar to natural human movements, eventually damaging the immersive property of VR games. To capitalize the immersive characteristics of VR gaming, a virtual environment that allows players to explore freely, an Open World setting, is deemed necessary by the Team.

With a rising concern on lagging survival skills and workmanship in younger generations, the Team decided to explore the possibility in using VR gaming as a tool to educate as well as preserve the losing skills.

The objective of the project is to create an open world virtual reality role playing game, with a tutorial stage which train the player certain skills related to survival or craftsmanship.

The project would need to let the player explore the surroundings without illogical constraints or straight guidance. The player should be able to at least describe the basic steps or knowledge necessary to replicate the in-game skills in real life.

To facilitate the open world exploration, a platform which allows the player to walk at the same spot would be produced as a side project.

The Team consists of four team members and their contribution are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>UID</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hui Sen Fung Felix</td>
<td>3035090327</td>
<td>Gesture recognition module development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Locomotion design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UI development</td>
</tr>
<tr>
<td>Leung Ho Ki</td>
<td>3035097210</td>
<td>Background story writing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stage design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skill research</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Platform development</td>
</tr>
<tr>
<td>Lau Chui Shan</td>
<td>3035097325</td>
<td>3D modeling &amp; animation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Player interaction scripting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AI scripting</td>
</tr>
<tr>
<td>Wai Yip Yin Calvin</td>
<td>2008993564</td>
<td>Game logic programming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Platform development</td>
</tr>
</tbody>
</table>
SECTION II - PROJECT BACKGROUND and LITERATURE REVIEW

1. Background

VR, by definition, is a computer simulated 3D environment, which provides realistic and immersive experience, controlled by player’s body [1]. In the past 3 years, Oculus Rift (2010 first prototype, Released in 2016), Google Cardboard (Released in 2014), PlayStation VR (Released in 2016), and HTC Vive (Released in 2016) came to the market, making VR gaming an actual reality to gamers.

Following the gaming consoles, virtual reality specific games are published for the market. Those games could be played on either mobile phones or consoles, and some of them are migration from traditional control scheme to VR control. Some companies saw the opportunity and manufacture other VR gaming related gears, including running platforms and backpacks.

One of the VR applications, other than gaming, is training. As to provide a safe environment for trainees, while still giving the experience they could acquire in real life scenario.

As a Hong Kong citizen, it is worth concerning that traditional and survival skills are not being equipped by younger generations. Survival skills, often only being taught in extra-curricular activities, are not always available to students, not to mention sources from certain media may be confusing or misleading in learning these skills.

Traditional skills and craftsmanship are skills that often taught by masters, while pupils need to spend a dedicated amount of time to perfect their basic skills before learning the true techniques. It is easy to find that under current education system, students may not have the time nor will to learn the skills they or their parents perceived as hard to earn a living. As the amount of pupils decreased, it is harder for skills to pass down the generation, and eventually lost. There exist some services that teaches people piano using AR technology, and with collaboration with VR training, the technology can act as an archiving tool for skills.
2. Previous Work

2.1 VR Games

At the time of writing, there are 1443 games that supports VR viewing or control, only 25 of them have a open world setting. After investigation, the majority of these games are driving games, instead of a first person game that the player can experience the gaming world themselves.

In the game collection, current VR games are mostly simulation games, allowing the player take the position of certain duty, and perform operations per request. Those games often limit the player in a relatively static geographical position, so as to adapt to the physical constraint of the gaming console.
Another type of VR games allows player move around the gaming world, with methods to be further discussed in Sub-session 2.3 VR Locomotion. Those games are mostly action games, allowing the player to perform certain acts in quick succession.

To conclude, current VR games are able to let player interact with the digital world effectively, but often limited their performance by the technical difficulty induced by the implementation method of consoles.

2.2 Open World Games

![Fig II.2.3 ARK: Survival Evolved, an open world game](image)

Open World games have long been in existence as a game genre. Providing a large and open environment to players, open world games are constructed as a non-linear gaming experience, and is much closer to the real world by design. Open World games have three special features: a seemingly limitless world map, non-linear gaming progression, and constraints that prevent players from breaking the game.

The map in open world games can be divided into two kinds: generated map and single world map.

![Fig II.2.4 Generated Map of Minecraft](image)
Generated map is the true limitless world, as the world is generated according to the player’s travel, with the only limit be the hardware of the gaming platform. This kind of map is often used in sandbox games, for example Minecraft.

Single world map is simply a very large map, with stages and zones all incorporated into a single map. This kind of map is often used in RPGs, as the map is often one of the storytelling element and allows the development team obtain a higher level of control over the whole game.

In Open World games, players are not required to follow a very straight route or steps to complete the game, as the destinations of players are not strictly limited by the map or stage design. One popular solution to this is using a quest system, which uses inter-linked tasks to create a multi-lined mission system. While having very high flexibility in terms of expansion and allows modification later in development, a huge quest system could be very hard to manage and falsely implemented, resulting to the existence of impossible tasks and unused tasks.

Constraints exists in basically every games, but Open World games have to be very cautious in setting those restrictions.

One typical restriction difficulty often exist at the border of the map, as some RPG maps are not truly limitless, players have to be actively discouraged to go beyond the map limit. Solutions that is very popular in those games included the invisible wall and warning text. Warning texts is usually displayed once the player cross the border and sometimes forcefully teleport the player back into the map, while invisible wall stop the player character at the border from moving further, despite there is clearly room to transverse, severely harming the immersiveness of the game.
2.3 VR Locomotion

Locomotion methods are critical in VR games that want to implement a larger map and allowing player to move around the map in order to perform certain acts. Current solutions to the matter involves three major means: teleportation, mimicking the moving action, and motion platform.

Teleportation is performed by letting the player signifies the desired location, either by pointing or placing a token, and then changing the player’s location to the destination.

![Locomotion by teleport](image1)

Based on hardware, there exist different ways the games handles the gesture for moving around the game, however those are most likely to be not very natural to the human body or weird looking.

![VR Motion Platform](image2)

Motion platforms let the player walk on them without worrying going away from the gaming area, and feel naturally while traversing the digital world. However, there have been reports that the harness those platforms used to constrain the player are not comfortable for long term gaming, and those platforms are too large to be used in domestic environment.
2.4 VR Training

VR has long been used for training for professionals with high risk jobs, including military and the aviation sector.

Initially in a form a simulated environment, specialized machines are manufactured to replicate the actual working scenario. Take pilots as an example, since hazardous scenario cannot be replicated in real life and it is not practical for trainees to search the world for such scenarios, a simulated cockpit is developed with machines that can mimic different flying scenario consistently.

There is a trend that VR training is moving into the service for the general public. There are VR trainings for car drivers provided in some driving schools in Hong Kong, lowering the anxiety and difficulty for first time drivers.
SECTION III - PROJECT SCOPE

1. Game

The project should include designing and developing a VR game that:

a. The Game should be able to run on Windows 7 or above operating platform, utilizing a mobile phone and a VR headset as the display for the player.

b. The Game should consists at least one skill training criteria, which is detailed and instructional enough for players to perform or describe the skill after playing the game.

c. The Game should provide an Open World setting for player’s exploration, with a quest system for maintaining the game progress.

d. The Game should provide logical barriers and constraints for preventing player from performing unwanted actions.

2. Gesture Recognition

The project should include designing and developing a gesture recognition system, controlling the VR game, that:

a. The System should be small scale enough for typical Hong Kong domestic use.

b. The System should be easy to set up and uses the player’s whole body for control.

c. The System should take gestures as close to natural human body movement as possible for control and locomotion.

3. Platform

The project could include designing and developing a motion platform that:

a. Being able to facilitate the motion capture of the Game.

b. Safe and Confortable enough for long term usage.
SECTION IV - PROJECT METHODOLOGY

1. Game Development

The Game Development was divided into 4 main phases: analysis, design, implementation, and testing. The 4 phases were carried out under the agile system development life cycle, as to facilitate the project development in the short period.

In the analysis phase, the skill to be taught was selected and a set of basic gestures was selected based on typical gaming requirements. The necessary procedure and techniques to properly perform the skill was then researched, and a set of skill specific gestures was selected accordingly. The overall theme of the game was also chosen based on the trend of currently available VR games.

In the design phase, the game background was designed. Followed by the visual design and program structure. The visual design shall align with the design theme chosen and the program structure shall be designed to facilitate multiple programmers working at once without sacrificing consistency and performance.

In the Implementation phase, each game object was modeled and placed in their respective group in the project hierarchy. Scripts were produced and structured before attaching to their respective game object. It is critical that programmers and 3D modelers keep a constant communication to satisfy each other’s need.

In the Testing phase, an internal test involving the Team only would be carried out, followed by an external test involving testers invited by the Team. The phase aimed to find out the bugs of the product as well as fine tuning the parameters for difficulty.

2. Gesture Recognition

Gesture Recognition first started by listing out the desired movements when the player perform a selected set of operations. Those movements were then analyzed and critical points were selected, which are the turning points of motions or extreme points of the motions.

Those critical points were then translated into joints positions, and implemented into the gesture recognition script by comparing the relative positions of the joints.

The suitable hardware was chosen by its availability to the Team, its capability for the task, and its suitability to the general public.
3. Platform

The motion Platform was created as a side project.

The extreme points of the player’s motion were defined and measured, and the design of the platform was derived. The development process underwent vigorous prototyping and revision. The design was planned to be able incorporate most motion detection products.

SECTION V - IMPLEMENTATION

![Diagram of Game Platform](image)

**1. Game Design**

The game design is divided into story design and the stage design.

**1.1 Story Design**

Since the Team selected survival skills are the skills to be taught, a survival sci-fi world with a post-apocalyptic scenario was selected as the background story, prompting the player to survive under extreme environment.
The name “Powerless” was chosen as the game’s title, as it was said that after a solar storm, most electronics and power generating facilities were damaged, cutting the whole world from electricity.

After years of repairing, the world regained limited amount of electricity generation, and is managed and used by the upper-class. Lower-class citizens only receive little amount of batteries for maintaining a living, while their upper-class counterpart walled themselves in and enjoy a luxurious life.

The player was an upper-class citizen who fight for the lower-class’ fair use of energy. As the result of the player’s rebellious act, the player was casted out of the walled city into the wilderness, where the player must fight for survival and rise up to revolutionize the energy distribution.

The story is designed to suit the open world setting by adopting the multi-line plot. Although the main story line was already designed, only the tutorial stage will be implemented for the current stage due to time constraint.

1.2 Stage Design

The stage design mainly surrounds the tutorial stage, as it would be the only quest to be implemented. The stage design treasures the open world setting and is not limiting the player’s action, but providing guidelines and optional goals.

The current stage design consists of 5 tasks: collecting resources, building shelter, starting fire, hunting, and bandaging. These tasks are selected by the Team as the basic survival skills.

The tasks were designed to follow a set of guidelines followed by their respective outcome. Such tasks are structured sequentially such that the player can follow them easily and logically.

2. Project Hierarchy

The project hierarchy is divided into three main parts by their functions: the Item Manager, the World, and the Game Master, inspired by the widely known website MVC framework. Such hierarchy is adapted to maintain a clear and manageable object system, minimizing the occurrence of redundant instances, while ensuring necessary object is present for game progression.
2.1 Item Manager

Put forward by the Team’s 3D modeler, the Item Manager organizes items by quests, and further divided into tasks. For the current installation, the game only consists of a single tutorial quest, which divides into three tasks, plus an additional task object containing the resulting objects after completion.

The main function of the Item Manager is to instantiate the game objects by their respective quests and tasks, manage the amount of items present in the game world, destroy any items that have been used.

Due to the open world setting, the game is progressed by quests, which required dynamic management of quest items, in order to prevent the occurrence of inadequate resources present in the world, rendering the progression of tasks impossible.

2.2 The World

The World is an entity containing the map and player, the most basic and critical game objects in the whole game. These two game objects are handled separated from items is due to their static characteristic, as the map (terrain) and the player are not dynamically generated and changed. This ensures the game is technically still playable even without the quest system, opening up to the implementation of planned feature, the Sandbox Mode.

The World acts as the starting point of the whole game, initiating the logical system upon awake. It also manages the player instance with UI element, and all terrain related instances, including trees and bushes.

2.3 Game Master

Game Master is the logical unit of the whole game, managing the progression of the quest system and the gesture detection system. Started upon the complete loading of the World object, the Game Master starts the Quest System.

The quest system default starts the tutorial stage at the current installation, the quest sub-system requests the instantiation of objects from Item Manager, detects whether the quest goals are met, and progress to the next step. After completing all goals, the quest system waits for the next trigger to start a new quest, where further quests are not implemented currently. When there is no active quest, the player is said to be in free roam mode, a near Sandbox Mode.

The gesture detection system is currently placed at the Game Master object, as it interacts with nearly all instances in the game. The gesture detection system takes the input of Kinect and translate the joint position into in game vectors, by comparing the positions of joints to each other, a gesture can be sensed and corresponding functions can be called.
3. Scripts

The scripting of the game span through the project hierarchy, and can be divided into four categories: Item Behaviour, Player Behaviour, and Game Logic. Some variables are exposed for fast difficulty tuning, and inputting parameters for generalized scripts.

3.1 Item Behaviour

Items in the game possess different roles including building materials, buildings, consumables, and tools. The scripts are therefore divided into their respective functions:

3.1.1 Object Handler

Object Handler describes the object’s type and other parameters, facilitating the operation of other scripts.

3.1.2 Pickables

The Pickables is an object class that signifies the object is able to be picked up by the player. The class is necessary for inventory management, since the player cannot logically pick up all materials, may it be too large or too heavy.
3.1.3 Building Handler

Building Handler is attached to a special piece of building material, namely Base Material, which act as a critical building block and handles the trigger for completing the building.

The Building Handler was initially built only for building the shelter, the script was then revised in code review and modified into a more general structure.

The Building Handler makes use of a dictionary for recording the required materials and their respective amount. The currently connected amounts are then updated whenever there is any collision or removal of materials.

Once all required materials are satisfied, a counter is initiated. Such counter is used to countdown the frames before instantiating the resulting structure, giving the player some window to cancel the operation, as well as encouraging a more delicate placement of materials.

When the counter finishes, all components will be destroyed, followed by the instantiation of the resulting construction, and finally destroying the base material, completing the living cycle of the material.

3.1.4 Skill Handler (Fire Handler)

The current version of the game consists only one skill handler, the fire starting skill.

The Fire Handler is also to be attached to a specific kind of material, namely Fire Board, the base material for fire starting.
Once the player is using a drill and a bow as instructed by the tutorial, the Fire Handler will then record the displacement of player’s hand, multiplying the distance with a multiplier and translating it into heat generated.

The Fire Handler records the current temperature of the Fire Board. If the temperature reaches certain points, including the smoke temperature and burn temperature, the related functions in Fire object is then called to create the fire.

To emulate the real life fire starting, the temperature of the Fire Board is set to be decreasing in an exponential rate, due to the temperature difference to surrounding environment. To start the fire, the player need to perform the act consistently and fast enough.

One down side of the current program design is that the script is only applicable to a specific skill and in certain method only, generalization of the script is prefered.

3.2 Player Behaviour

The Player behaviour scripts handles all interaction player and in-game character, as well as the character’s statistics. The scripts included Player Statistics Handler, Gesture Recognition, Inventory Management, and User Interface Handler.

3.2.1 Player Statistics Handler

The survival design of the game means the player needs to deal with real life needs, including hunger and thirst. The Player Statistics Handler manages the health points, level of hunger and thirst, and current body condition.

With a constantly depleting hunger and thirst level, the game prompt the player seek stable food source and water supplements. When the player’s hunger and thirst levels drop through certain points, the player receive a negative body condition, with first level being hungry and thirsty, proceed into starvation and dehydration, and finally death.

When the player is in a negative body condition, the player would have lower moving speed, smaller strength, and depleting health.

3.2.2 Gesture Recognition

Gesture Recognition scripts can be divided into two parts: Kinect Adapter, and Gesture Control.

The Kinect Adapter translates the joint position data from Kinect Sensor into in-game vectors. The adapter will record the data once per frame and the Gesture Control is then called.

The Gesture Control detects the relative position of joints and using the information to determine if certain move or operation is being performed by the player. Once a certain gesture is recognized, the respective function is called.
There is also public functions that export certain joints’ current positions, which are then used in some external calculation, one example would be the Fire Handler, which uses the hand positions for determining the temperature created by the player. Such construction could lower the computing load since the function is only called when needed, and the determination step of whether to call the function or not is handled by another script, which can use trigger instead of checking per frame.

3.2.3 Inventory Management

The player can store any pickables into Inventory, a virtual backpack that can be accessed by placing player’s hand to the back.

The Inventory script consists of an arraylist, which records all items in player’s possession. The script provides basic functions including open and close the inventory window, and selection methods. Working with Gesture Recognition, the player can swipe through the inventory using hands gesture. After the player selects the wanted item, the script attaches the item to player’s hand, making it available to the player.

It is worth noting that items displayed in the World has a variable indicating the amount of a single entity, meaning if the player picks up one entity of certain object, the player would automatically gain a certain amount of units of that object. Such design is to reduce the object count when there is a larger world with pickables existing everywhere in the map.

When the player picks up a game object, the object is automatically transferred to the backpack, with the Inventory Management Script recording to total unit of items.

3.2.4 User Interface Handler

Since the player statistics are quantified, the player would need certain ways to access current body condition.

The User Interface Handler collects the data from Player Statistics Handler and displays in a easily accessible fashion. The display may be turned off to create a more realistic feel to the game, with a more subtle way to display any alarming conditions.

The Handler also handles the display of Inventory and the animation involved. The design of such display would be mentioned in the design section.

Current quest goals and instructions are also displayed to direct the player.
3.3 Game Logic

Game Logic scripts manages the progression of the game. The scripts including Game Master and Game Controller.

The Game Master is the highest level script, and is planned to control the game mode of the whole game. While the current version only consists of the quest mode, a free playing mode, the Sandbox mode is planned for future updates. Currently, the Game Master will start the Game Controller upon awake.

The Game Controller is a quest controller, which currently starts the tutorial quest by default. The Game Controller manages which tasks is currently in progress and all other potential quests available to the player. Once all task goals are finished, may it be finishing the construction of a building, able to perform certain act, the task is said to be finished, and the related items will be processed accordingly.

Such design enables the game to include future changes in game mode as well as increased amount of quests, without affecting the original quests.

4. Visual Design

The visual design was determined to be realistic to give a more immersive experience. The design can be divided into player character, UI, and the environment.

4.1 Player Character
Fig. V.4.1 Player Character Model by Lau Chui Shan

The player character was designed with a worker-like appearance, as to signify the lower-class status of the player.

The 3D model was produced using MakeHuman, and exported into Unity 3D. Yet it was found that the head of the model would prevent the character from following the player’s action, hence a further modification was carried out to remove the head part, while maintaining the skeleton for camera control.

4.2 UI

The UI implemented at the current stage are the inventory menu and player condition.

The inventory menu is designed to display maximum 3 items at the same time, with navigation by swiping left or right. The displayed item is in the form of an image, which is captured in the development stage, with the item’s name displayed at the bottom of the image. General instructions on using the inventory menu are also displayed, in the form of text.

Fig. V.4.2 Inventory Menu by Lau Chui Shan

Fig. V.4.3 The HUD for player’s statistics and quest by Hui Sen Fung
The player condition is an always displaying as a HUD, with 4 player statistics, namely health, hunger, thirst, and injury level. The statistics are displayed at the side of the screen to prevent obstruction to the player’s view, and the display color would change on the condition of the player. The HUD also displays the current quest’s goals, as the instruction for the player.

4.3 Environment
The environment involves the map and all items present in the world.

4.3.1 The Map

Based on the tutorial requirements, the map requires a source for wood and another for water, and was designed to be a forest and river respectively. Since this is a survival scenario and a tutorial stage, an island is chosen to be the map, as the player was casted out into the wilderness.

The Map was built using the terrain gameobject in Unity3D, after assets on the assets store were deemed not suitable for the project. Since the Map size is limited, constraints have to be implemented for discourage the player from going beyond the boundary. As a result, the gaming area is surrounded by cliffs and slopes, while the player is discourage to go into the open water as it is highly polluted.

Optical illusions, including changing the size of objects according to distance, are implemented to trick the player into thinking the map is bigger than designed. Functional high-points like rocks are placed for escaping from hostile animals.

4.3.2 Items

In order to save time in development, existing resources are used for the in-game items. While the resources can satisfy the project’s needs, certain special items, including constructions like the shelter, have to be tailor made.
Usually buildings would be divided into steps when progressing through the quest, to reduce the object count and give the player a more consistent performance, the constructions are built into intermediate steps.

5. Gesture Recognition

Apart from the scripts mentioned above, the gesture recognition consists of the hardware part.

The Team was originally planned to use inertial full-body position sensors for the motion capture for the project, longing for their wireless and accurate features. The Perception Neuron by Noitom was chosen after comparing 3 other alternatives. However, it was found that a system necessary for the Team would cost slightly more than HKD 4000, which exceeds the Team’s budget.

As an alternative, the Team borrowed a Kinect sensor by Microsoft from the Department of Computer Science, the University of Hong Kong, for the motion capture. The sensor strike a lower performance when comparing to inertial sensors, yet the low cost outweighs the disadvantages.

The current version allows the player changes the facing direction of the in-game character with a mouse click, as to counter the Kinect’s incapability to differentiate the front side and back side of the player, this is planned to be changed and solved by the platform design mentioned below.
6. Platform

Before the Platform was designed, an experiment and measurement was carried out to figure out the correct size.

Since the Team chose to use the Kinect sensor, the minimum detection range has to be taken into consideration.

![Fig. V.6.1. Testers Performing selected postures](image)

![Fig V.6.2. Kinect Detection Distance [12]](image)

In the experiment, the Team concludes that every individual has unique posture even when performing the same action. Taking personal height into consideration and the experiment conclusion, the Team estimated a range for allowed actions to be performed, and the platform is designed using the data.
The design of the Platform consists of two poles attached to a circular track. The two poles are kept in opposite side, one of the poles will hold the player at the center of the platform using a belt to the waist (the “Player pole”), while another one will be supporting the Kinect and maintaining the Kinect facing the player (the”Kinect pole”).

The Player pole would fix the player’s horizontal position relative to the platform, while allowing vertical movements. The Kinect pole on the other hand, would keep the Kinect at the same level and maintaining a minimum distance from the player for best motion detection.
SECTION VI - CONCLUSION and FUTURE WORKS

1. Conclusion

By completing the alpha stage of the game production, the Team can conclude that skills can be digitized and preserved in the form of VR games, allowing players not only viewing past masters performing the techniques, but receive hand-to-hand tutorial from the masters, even if the skills are long lost in the real world.

However, the learning outcome may not be satisfactory for every individual. This could be the result of the fact that VR games cannot handle the questions posted by players if the questions are not anticipated, or the learning style does not match with the player’s.

It can also be observed that while computer vision based motion capture is way cheaper, initial position sensor is more suitable for domestic use in Hong Kong, thanks to their small required operation space. It is perceived if the cost of the sensors are lowered in the future, VR gaming would be made available to more players.

Open World setting for VR games is hindered by the technical constraints of the gaming consoles and platforms. However this does not change the fact that Open World gives a more satisfying gaming experience than their more static counterpart.

Reflection on the development process points out a structured object hierarchy and program structure is critical to the success and efficiency of the project.

2. Future Works

It is proposed not only include a single kind of skills (survival skills only in this case), but a range of skills in the game, if Open World setting is made available by technical advance. As Open World setting allows a more flexible story-telling method, developers could take advantage of the characteristics and expend the value of the game.

As suggested from the conclusion, exploration of different education style used in the game is prefered, as different individuals have different learning style. While teaching in institutes could sometime be restricted in certain style, gaming can take a more flexible approach and let the player choose the learning method, as the necessary information is already recorded in the game.

While survival skills focus on knowledge and improvisation, traditional skills may rely more on feelings and experience, which are not fully delivered in current installations. There exist haptic feedback body armor that let player sense where they have been hit, it would be beneficial if a more precise and delicate version is developed for the hands, such that players can have a more complete experience to their game.
References


