Intermediate Report

Project Title: A Motion Tracking Game Based On VR

Student:
Ng Cheuk Hung (3035281093)

Supervisor:
Dr. T.W. Chim

Date of Submission: 20/01/2019
Abstract

Virtual reality with motion tracking technology has been utilized in different industries around the world recently, from engineering simulation to gaming industry. Retrieved motion can be reconstructed inside the virtual world built by the software. It can then be used for simulating the interaction between users and computer-generated virtual world. However, the rudimentary implementation of such simulated interaction in video games can be improved further to elevate the whole gaming experience. In this paper, a new model named motion-driven player-game interaction is introduced as a conceptual solution to improve current virtual reality games with motion tracking devices. A detailed motion tracking algorithm will be designed and a demonstrative virtual reality game will be developed by the end of the project. HTC VIVE is chosen as the core virtual reality and motion tracking device in this project. The game is planned to be built in Unity Engine with the HTC VIVE. Currently, a motion tracking and pattern recognition algorithm is designed in order to efficiently detect and distinguish different postures and movements of the users and trigger the corresponding game events. The development of the demonstrative game in early January, and the project is slightly behind the schedule. Accuracy of the translated motion data and performance of the designed motion tracking algorithm are limited, but they could be improved with better algorithm implementation in the future work.
Contents

Abstract 1
List of Figures 3

1 Introduction 4

2 Background 6

3 Objective 9

4 Scope 10

5 Methodology 11
   5.1 Virtual Reality Gear and Motion Tracking Devices 11
   5.2 Game Engine 11
   5.3 Software Development Kit 11
   5.4 Agile Methodology 12

6 Discussion of Results 13
   6.1 Design on Motion Tracking Algorithm 13
   6.2 Game Design 16

7 Limitations and Challenges 17
   7.1 Inaccuracy of Motion Tracking Devices 17
   7.2 Overhead on Constant Motion Interpretation 17
   7.3 Orientational Data for Motion Tracking Algorithm 17

8 Future Plan 18

9 Conclusion 19

References 20

Appendix I - Project Schedule in Project Plan 21
List of Figures

1  Screenshot of player before turning over his hand  7
2  Screenshot of player after turning over his hand  7
3  Flow chart of motion tracking algorithm  13
1 Introduction

Virtual reality games have recently become more popular in the gaming industry. Steuer’s article pointed out that virtual reality is a technology to provide user an environment for telepresence experience [1]. Different game development companies and studios have introduced motion tracking technology into their virtual reality games and this has been a hot topic in the gaming industry. The journal article from Rosenberg and Page described motion tracking device as a type of computer input devices for converting human motion into data which can be interpreted by computers [2]. With the use of motion tracking devices, virtual reality game can bring evolution to the video game industry.

Currently, most of the virtual reality games make use of the motion tracking technology to solely capture the human motion and reproduce the body movements on in-game characters. Therefore, this project aims to expand the usage of motion tracking technology on virtual reality games besides the reconstruction of player’s body movements inside the game. There are a lot of information the game can retrieved from the body movements. Body motion can be interpreted as a sequence of human body postures to complete a set of particular tasks [2]. The motion sequences and patterns of the players can be used as a type of data input to control different elements and accomplish different tasks inside the game. As a result, the interaction between the players and the games is enhanced by the enlarged and improved implementation of motion tracking technology on the virtual reality games. Players can then have a better control on different game objects in the virtual reality world with the enhanced player-game interactions.

A motion tracking game based on virtual reality will be delivered in the final stage of this project in order to demonstrate how to utilize motion tracking devices in virtual reality game by interpreting the motion sequences and patterns to greatly improve the player-game interactions. This concept of player-game interaction based on motion tracking will be named as motion-driven player-game interaction in this project. An efficient algorithm for transforming body motion into in-game commands is introduced as well. The current progress of the project is slower than expected. Some motion estimation algorithm and related studies are investigated to improve the understanding on the concept of motion tracking techniques. A motion pattern recognition algorithm is then designed for
distinguishing different body movements given a set of predefined motion patterns in the database. However, the actual implementation of the game has just started in early January.

The remainder of this paper proceeds as follow. In chapter 2, the background of the motion tracking algorithm and the motivation of this project will be introduced. In chapter 3, the objectives of the project will be described clearly. The project scope will be included in chapter 4. It will give some information on the project focus and define the works to be done. Chapter 5 of this paper will describe the methodologies of the project, from the game engine to the motion tracking devices. After that, chapter 6 will give detailed explanation on current findings on the motion tracking algorithm. The limitation and expected challenges of the implementation of the designed algorithm will be included in chapter 7. The paper will end with a short conclusion in chapter 8.
2 Background

In this section, the background of motion tracking game based on virtual reality and the motivation of this project will be explained in detail. A clear introduction to the motion-driven player-game interaction with examples will be included.

Virtual reality technology in gaming industry brings a new level of experiences to gamers. One of the main goals of video games is to provide new experience to the players. Virtual reality can change users’ perception to their surrounding environment [1]. While the players enjoy their new experiences inside the virtual world simulated by the virtual reality games, there are needs of interaction between the players and the game objects. Motion tracking devices is then introduced as a method for players to interact with the game.

As stated in the article from Smith, a modern virtual reality system contains a game station composed by input and output devices, including a motion tracking device for sending output signals by detecting the movements of the user [3]. Some of these motion tracking devices can precisely capture the motion of the users and send the corresponding signal to the computer. The game can then interpret the received signal and simulate the interaction between the player and related game objects. Therefore, players can interact with the game with their body movements now while players in the past can only interact with the game with traditional computer input devices, like keyboard, mouse and console controller.

Motion tracking games based on virtual reality have become the spotlight among the gaming industry in the past few years. All the top-tier virtual reality games in the Best of 2017 Virtual Reality Rewards from Steam make use of the motion tracking controller [4]. Steam is one of the biggest digital distribution platform for video games in the world. Its statistical data and attention on virtual reality games shows that the market of virtual reality games has grown quickly in recent years and game development professionals start using motion tracking technology in their virtual reality games.

However, most of these virtual reality games solely utilize the motion tracking controller to capture the hand movements and reconstruct the motions inside the game. This project suggests that more advanced utilization of motion tracking devices can be used in the game
implementation for better player-game interaction improvement. Player should be able to interact with the user interface or even control different game objects through body motions.

Figure 1: A screenshot of player before turning over his left hand

Figure 2: A screenshot of player after turning over his left hand to open a menu

There are some examples to clearly explain the suggested advanced implementation of motion tracking devices and the concept of motion-driven player-game interaction. Leap motion is a company producing hand motion tracking devices in the gaming market. They published some videos showcasing their hand motion tracking devices (See Figure 1 and 2). In the selected figures, player can turn over his left hand to open a menu. This is an example of utilizing player motion to interact with the user interface. The player may then turn over his right hand for other game events which depends on the actual game implementation. Player motion becomes the “language” between players and the game. The game can interpret the player motion and understand what game events should be raised correspondingly. This paper describes this type of player-game interaction as motion-driven player-game interaction. Its name implies that this type of human-machine interaction starts from the user motion as the input.

In short, the motivation of this project is to improve the current motion tracking technology implementation on virtual reality games based on the concept of motion-driven player-game
interaction. The project aims to design an efficient algorithm for implementing this motion-driven player-game interaction on virtual reality games.
3 Objective

The following section describe the objective of this project clearly. There are two main objectives in the project, namely the algorithm design and the game demonstration.

This project ultimately aims to enhance the implementation of motion tracking devices on virtual reality games to provide a better player-game interaction. This is because most of the current implementation of motion tracking technology on games based on virtual reality is too simple and straightforward. For instance, hand movements captured by the hand motion tracking devices are usually merely used to reproduce the tracked motions on an in-game character. However, more advanced utilization of the body motion data can be implemented to the game, such as triggering different in-game commands and effects through different body motion sequences and patterns.

Therefore, there are several sub-tasks to be completed in this project. Firstly, an efficient algorithm will be introduced for transforming motion sequences and patterns into a type of information to be interpreted and utilized by the game, like mapping motion into an abstract matrix form. Then, a motion tracking game based on virtual reality will be made to demonstrate how virtual reality games can make use of those useful data information generated by motion tracking devices on different game elements to provide a better player-game interaction. For example, drawing a circle with the motion tracking device can open a setting menu while most of the current virtual reality games require players to watch at the setting menu icon for a few seconds to open the menu.

In order to demonstrate how the motion tracking technology can be used on video games, the game product will be a virtual-reality fighting game. The motion of the player will be used on fist and weapon controls, and also different specific motion patterns will be used to trigger different abilities to fight against the enemies. For instance, when a player pretends to shoot an arrow, the game can detect the motion and correspondingly spawn a moving arrow according to the player’s direction.
4 Scope

In this section, the scope of the project will be described. This part of the paper will define clearly the focus of the project and the works will be done in the next few months. Some out-of-scope tasks are included as examples to discuss the project scope.

The focus of this project is to study how to use advanced implementation of motion tracking devices in a virtual reality game to improve player-game interaction. Therefore, the algorithm study is focused on how to transform motion information to simple movement sequences and patterns, like a movement sequence as LEFT $\rightarrow$ TOP $\rightarrow$ RIGHT. Then these sequences and patterns are classified and used to trigger corresponding predefined in-game commands. Apart from the algorithm study, a game will be made as a practical application to illustrate the whole process of motion-driven player-game interaction. The generalized process is proposed to start from motion capturing with motion tracking devices to interpretation and simplification of the input motion data, and finally to trigger customized and predefined in-game events.

Therefore, the algorithm study does not aim to improve motion tracking devices or motion tracking technology but to propose a model to utilize the captured data efficiently on a virtual reality game in a larger extent. The project will not solve the problem of precision and accuracy of the motion tracking devices and presume that the used motion tracking device is capable of tracking the body movement of the players precisely and accurately. Although a game will be delivered in the final stage of the project to demonstrate the proposed motion-driven player-game interaction implementation, it will not be comprehensive enough to be comparable with the commercial games. The delivered game is supposed to showcase the advanced implementation of motion tracking technology on virtual reality games and its effects on raising the game enjoyability, therefore the other elements of the delivered game, such as storyline and aesthetics, are not weighed as much as the aspects of technology and mechanism.
5 Methodology

In this section, the methodology in this project is discussed. Different tools used to implement the project are introduced in the following part, including the virtual reality gear and game engine for implementing the virtual reality game.

5.1 Virtual Reality Gear and Motion Tracking Devices

The virtual reality gear chosen for this project is HTC VIVE. HTC VIVE is a virtual reality gear developed by HTC and Valve Corporation. It consists of a headset and a controller. The controller can be served as a simple hand motion tracking device. The reason why it is chosen for this project is that HTC VIVE has a high accuracy on position and orientation calculation and its end-to-end latency is only 22ms [5]. It is also well supported by the game engine and software development kit to be discussed below. Therefore, the development of the delivered game with HTC VIVE is more convenient and has less risks.

5.2 Game Engine

The game engine used to implement the delivered game is Unity Engine. Unity Engine is one of the most popular game engine for creating three-dimensional games. It provides a set of scripting application programming interfaces in C# for developers to implement custom game logic. Unity Engine has a great support on virtual reality game development and development with HTC VIVE. It provides an input system for mapping the input from motion tracking devices. Moreover, C# is an object-oriented programming language, so it is easier to implement game logic since it treats the elements inside the game as an object. The interaction between the objects can be modified to illustrate the motion-driven player-game interaction. Therefore, Unity Engine is chosen for this project.

5.3 Software Development Kit

SteamVR is chosen as the software development kit and the run-time environment of the game when implementing the delivered game. SteamVR consists of OpenVR, a set of application programming interfaces for the virtual reality game development with the feature of independent update for supporting hardware updates according to its documentation [6], and a run-time environment for virtual reality games on PC. SteamVR is required as a way for mapping device input to game-usable data in Unity Engine when the game is developed.
with HTC VIVE. Therefore, this software development kit is included in the development tools in this project.

5.4 Agile Methodology
This project follows the agile methodology. In order to perform the iterative development, a preliminary game will be development after the algorithm study to test the feasibility of motion-driven player-game interaction. After the successful build of the test game, an improved game product will be built on top of the preliminary game by adding more gameplay elements and more advanced game logic implementation. For each game build above, the game will be tested by some voluntary gamers for the feedback on the enjoyability of motion-driven player-game interaction and the stability of the delivered game. The game is then further improved and enhanced according to the feedback. Therefore, the project quality is ensured by following the agile methodology.
6 Discussion of Results

In the following section, the current findings and studies are discussed. Most of the current results in the project is about the algorithm study and design.

6.1 Design on Motion Tracking Algorithm

![Flow chart of motion tracking algorithm](image)

**Figure 3:** A flow chart of the motion tracking algorithm

In order to bring the motion-driven player-game interaction to the virtual reality game, a motion tracking and pattern matching pipeline is designed (See Figure 3). The pipeline consists of four main part, namely the point tracking process, feature extraction process, pattern recognition process and the thresholding process. Implementing the motion tracking mechanism using this pipeline can help the game detect and distinguish different motion sequences and patterns and therefore raise the corresponding game events. The whole pipeline will be completed within a short period of time, such as 0.1 second, and then repeat again. Therefore, an efficient and fast algorithm is very important for this pipeline. In short, the program measures the player motion and updates the position of the controller in the game. After that, the new positional data is used to update the motion sequences in the previous time intervals. Then, the extracted motion sequences are compared with the predefined motion patterns. A matching score is given after each comparison. For the most perfectly matched one, if the matching score is better than the threshold value, then the motion sequence is successfully matched with the pattern and therefore the corresponding
game events will be raised. Each part of the algorithm will be explained in details in the following part.

6.1.1 Point Tracking Process

The starting point of the algorithm is to measure the positional change of the player or the controller in this case. Motion can be defined as a sequence of changing position. In the Unity Engine, position is represented as a transform object storing the coordinates in a three dimensional space. This process can be done by the API provided by SteamVR. The API will received the data from HTC VIVE controller and update the values of the 3D coordinates. For instance, the position of the controller in \( t = 0s \) is \((2, 1, 0)\) and after the player moves, the new position of the controller in \( t = 0.1s \) is \((3, 1, 1)\).

6.1.2 Feature Extraction Process

The next part, which is also one of the most important part of the algorithm, is how to extract the motion and stored them in a form to be used for pattern matching in the later stage. The naive solution is to directly use the data from the point tracking process, the values of 3D coordinate system. The program stored a sequence of points in 3D coordinate system in the past one second and treat it as a representation of motion. However, this method has several problems. For instance, human motion is imperfect, therefore using exact position of the controller will make it difficult to match with predefined motion patterns because it is guarantee there will be slight positional differences between the tracked points from controller and the point in the predefined motion patterns. The speed of the controller may also result in large positional differences for some points in the motion sequences. For instance, player A and B want to make the same linear motion with \((0, 0, 0) \rightarrow (10, 0, 0) \rightarrow (20, 0, 0)\). Player A moves the controller faster and therefore perform a motion \((0, 0, 0) \rightarrow (15, 0, 0) \rightarrow (20, 0, 0)\) while player B moves slower and perform a motion \((0, 0, 0) \rightarrow (5, 0, 0) \rightarrow (20, 0, 0)\). Both of them are moving the same direction but the speed of their movement results in the difference of the second points. It is very hard to distinguish this type of situation using the point-to-point approach.

From the previous point-to-point approach, there are two properties can be utilized to design a better method to extract the motion features. The first one is imperfect human motion. The motion does not need to represent the human motion precisely since the algorithm do not need precise motion comparison. The second one is the different speed of the human
movements. In order to solve the problem, instead of storing the exact intermediate point of the controller in the motion sequences, storing which area the controller has visited should be a better approach. Therefore, fixed-size 3D matrix can be used to represent the motion for the efficient purpose.

The 3D matrix represents the space around the players. The values in the fixed-size 3D matrix will be set to zero by default and they will increase by a increasing amount if the controller is located on the corresponding cell in the 3D matrix. For instance, if the program stores motion sequences for the last 1 second and update the position of the controller per 0.1 second. For the first 0.1 second, the program add 1 to the cell where the controller is located. For the second 0.1 second, the program then add 2 to the cell where the controller is located. Therefore, the cell with lowest non-zero cell represents the oldest position of the controller in the first 0.1 second and the cell with highest non-zero cell represents the newest position of the controller in the last 0.1 second.

6.1.3 Pattern Recognition Process
Since the fixed-size 3D matrix is used to represent motion in this algorithm, the predefined motion should also be the same format for easy comparison. If the matrix for tracked motion is the same as the predefined matrix, then the motion is obviously matched perfectly. However, since the human motion is imperfect, adjacent cells should also be considered for each cell comparison. For each non-zero-value cells in the predefined matrix, if the value of the corresponding cell in the tracked motion matrix is zero, mismatching score is added by one and adjacent cells should be then used for matching too. If the adjacent cells are also zero, the motion is then said to be mismatch and another motion pattern should be used for the matching.

6.1.4 Thresholding Process
In the previous part, the mismatching score is added by one for each mismatch. Therefore, the matching score is zero for a perfect matching. A high mismatching score means more difference between the predefined motion pattern and the tracked motion sequences. A highly dissimilar pair of motion should be dropped, therefore the thresholding process is necessary. If the mismatching score is higher than the developer-defined threshold value, then the pair of motion should be considered as mismatched.
6.2 Game Design

By the end of the project, a virtual-reality fighting game will be made to demonstrate how can the motion-driven player-game interaction be used to improve the player experience to a greater extent. The game will be a single-player game for simplicity. The player will control a character and find with different enemies. The body motion of the player will be used to control fist and weapon, and also different specific motion patterns will be used to trigger different abilities to fight against the enemies. For instance, when a player pretends to draw a sword, the game can detect the motion and correspondingly spawn a sword in the player’s hand.
7 Limitations and Challenges

The following section of this paper will give some limitations and expected challenges in the upcoming stages of the project. Most of them are related to the accuracy and performance issues of the motion tracking process.

7.1 Inaccuracy of Motion Tracking Devices
The project presumes that the motion tracking devices can precisely and accurately capture the motion and send the correspond the signals to the computer. If the motion tracking devices fails to capture the motion correctly and send the incorrect data input, then the algorithm may output an undesirable in-game command. Since this is a hardware limitation and this project does not aim to solve this kind of problem, the inaccuracy of motion tracking devices may affect the game adversely independent of the algorithm.

7.2 Overhead on Constant Motion Interpretation
The constant interpretation of the raw motion data to the useful motion sequences and patterns may require a large amount of computational power and lower the performance of the game. This is because previous motions need to be stored for calculating the whole motion sequences and calculation of motion sequence and in-game command mapping is performed constantly upon motion detection. The algorithm implementation needs to be optimized to lower the negative impacts of constant motion interpretation on performance issues.

7.3 Orientational Data for Motion Tracking Algorithm
The current motion tracking algorithm does not consider orientational data as part of the motion. It is because it will be challenging to design a algorithm to recognise both the positional and orientational pattern. It will still requires a lot of computational power even if an algorithm is designed for the pattern matching on orientational change by motion. The performance of the game will be affected adversely. Therefore, this project currently has no will to design such a motion tracking algorithm for orientational motion pattern recognition.
Future Plan

In this section, the project schedule will be examined and the future plan will be also discussed.

The current progress of the project is slower than expected (See Appendix I). According to the project plan, all the algorithm study and design should be finished by the end of November 2018. However, there are still some parts in the motion tracking algorithm can be further improved, such as the pattern recognition process. For instance, the neighbor cell comparison should be explained in details on why is it necessary and how to ensure its correctness.

Therefore, the next task of the project is to finish the algorithm design completely. In the next few months, more effort will be put on the game implementation to ensure the game can be finished on time.
Conclusion

This paper introduces a new concept named motion-driven player-game interaction. Motion-driven player-game interaction is a type of player-game interaction which is initiated by the motion of the players. In order to achieve this kind of interaction, a motion tracking and translation algorithm is proposed for effective implementation. A demonstrative game will then be implemented by the end of the project to show the effectiveness on how motion-driven player-game interaction can improve the gaming experience. Currently, a motion tracking algorithm and its pipeline is designed. The implementation of the demonstrative game has just started with the completed algorithm. The motion-driven player-game interaction might improve the future implementation of virtual reality games. Players can have more controls on the game and more interaction with the games and other players, therefore the proposed type of interaction can be a solution to the problem of rudimentary gameplay design of the existing virtual reality games. However, the accuracy of the motion tracking algorithm is limited. Failed motion pattern matchings may lead to the frustration of some players since the player may find it difficult to interact with the game smoothly if the player need to perform the same motion again and again to match the pattern precisely. Therefore, the first future work is to investigate if there are any ways to further improve the precision of motion tracking algorithm in every aspect for giving a smooth player-game interaction. This task should be completed as soon as possible. The improvement of hardware for virtual reality technology may also help if a new and better virtual reality gear is released in the future.
References


### Appendix I - Project schedule in Project Plan - Page 18

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 September 2018</td>
<td>Deliverables of Phase 1 (Inception)</td>
</tr>
<tr>
<td></td>
<td>- Detailed Project Plan</td>
</tr>
<tr>
<td></td>
<td>- Project Web Page</td>
</tr>
<tr>
<td>October 2018</td>
<td>Research on Virtual Reality and Motion Tracking Devices</td>
</tr>
<tr>
<td>November 2018</td>
<td>Algorithm Study</td>
</tr>
<tr>
<td>December 2018 - January 2019</td>
<td>Development of Demo Game</td>
</tr>
<tr>
<td>7 - 11 January 2019</td>
<td>First Presentation</td>
</tr>
<tr>
<td>20 January 2019</td>
<td>Deliverables of Phase 2 (Elaboration)</td>
</tr>
<tr>
<td></td>
<td>- Preliminary implementation</td>
</tr>
<tr>
<td></td>
<td>- Detailed interim report</td>
</tr>
<tr>
<td>February 2019 - April 2019</td>
<td>Development on Deliverable Game</td>
</tr>
<tr>
<td>14 April 2019</td>
<td>Deliverables of Phase 3 (Construction)</td>
</tr>
<tr>
<td></td>
<td>- Finalized tested implementation</td>
</tr>
<tr>
<td></td>
<td>- Final report</td>
</tr>
<tr>
<td>15 - 19 April 2019</td>
<td>Final Presentation</td>
</tr>
<tr>
<td>29 April 2019</td>
<td>Project Exhibition</td>
</tr>
</tbody>
</table>