User Studies of a Multiplayer First Person Shooting Game with Tangible and Physical Interaction

ZhiYing Zhou, Jefry Tedjokusumo, Stefan Winkler, and Bingbing Ni

Interactive Multimedia Lab
Department of Electrical and Computer Engineering
National University of Singapore (NUS)
Singapore 117576
{elezzy,jefry,winkler,g0501096}@nus.edu.sg

Abstract. In this paper, we present a new immersive first-person shooting (FPS) game. Our system provides an intuitive way for the users to interact with the virtual world by physically moving around the real world and aiming freely with tangible objects. This encourages physical interaction between the players as they compete or collaborate with each other.

Key words: First Person Shooter, Virtual Reality, Augmented Reality, Physical Interaction, Tangible User Interface.

1 Introduction

First Person Shooting (FPS) games are a popular computer game genre. FPS games require a high accuracy of aiming, which is currently provided by the mouse [1]. This traditional method lacks physical and social interaction.

In this paper we present a new implementation of multiplayer FPS games. In our system, the players wear a Head Mounted Display (HMD) with head tracking, and carry a wand in their hand that is also tracked (see Figure 1). With these tracking devices we change the way multiplayer FPS games are played. The player's position and viewing direction are tracked by the head tracker. The wand's orientation corresponds to the gun aiming orientation in the game. As a result, shooting and viewing directions can be different in our system.

The game is played in a rectangular room without obstacles (see Figure 1). We place game items (health, armor, weapon, bullets) in the corners of the room. The player must physically walk to these locations to collect the items. The game style is "death match", which means the players shoot each other until their opponent's health reaches zero. The winner is the one with the most kills. The game style can be easily changed to support more than two players, so there can also be a collaboration component, e.g. battle between two groups.

The paper is structured as follows. In Section 2 we introduce previous related work on FPS games. Section 3 describes our system setup. Section 4 discusses

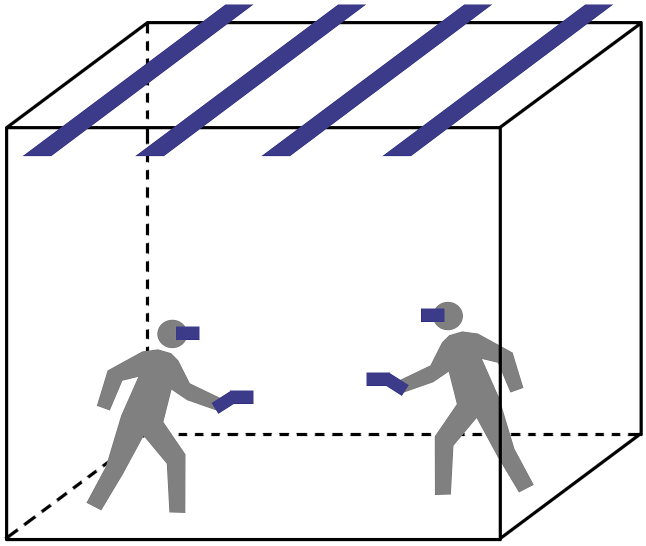

Fig. 1. Game environment.

the implementation of our system. In Section 5 we evaluate our system by means of a user study. In section 6 we conclude our paper and discuss possible future work.

2 Related Work

CAVE Quake [2] provided a great tangible user interface for FPS. CAVE is a 10x10x10 foot "cube" with images projected onto 3 walls and the floor. The player stands in the center, and the CAVE renders the virtual world in front, left, right, and bottom of the player. The player aims the gun using his hand as in the real world. The player typically uses a joystick to move around the virtual world. CaveUT [3] is a similar system; however, it is built using low-priced hardware equipment (around US$ 25.000) compared to CAVE Quake which requires million dollar equipment. Our system differs in the way virtual world is presented. Instead of using projection on walls, we attach a tracking device to the HMD so that the user receives updated first-person views whenever he moves.

ARQuake [4] is a single-player mixed-reality FPS. The player in ARQuake has the freedom to move around the world, but the gun aiming is limited to the center of the view of the Head Mounted Display (HMD), which means the aiming is done by the player's head rather than the player's hand. This is unintuitive, and it is difficult to aim accurately using the head while evading enemies. The game was initially designed as a single player game, so it has no interactivity with other human player. Though this game can be extended to support multiple players, the tracking systems that are used (GPS and markers) are meant for outdoors. Therefore, ARQuake may have accuracy problems to determine whether a bullet shot from one player has actually hit another player. Human Pacman [5] is another example of an outdoor augmented reality game. However, as it is also using GPS for tracking, it suffers from the same accuracy problem as ARQuake. Our system (InterSense IS900) is an indoor system with much higher tracking precision, which is essential for FPS games.

Touch-Space [6] is an indoor mixed-reality game that is situated in a room-size space, using a high accuracy ultrasonic tracking system from InterSense similar to ours. It has 3 stages: physical land exploration, virtual land exploration, and virtual castle exploration. The stages range from augmented reality to virtual reality. Our system differs from Touch-Space in that we add more physical interaction components, for example, collecting armor by moving close to virtual armor boxes, or jumping to avoid bullets. Furthermore, we implement a popular FPS game which involves intensive competition between users, while the tasks in Touch-Space require collaboration.

Beckhaus et al. [1] proposed ChairIO and a Game Gun as new control devices for FPS games. The ChairIO is basically a chair that tilts. The tilt is used to control the movement of the virtual character in the game (forward, backward, left, and right). The ChairIO also support jumping (by bouncing from the chair). However, the user is constrained to sitting on the chair; hence physical movements are very limited. Our system offers free body movements by using wireless trackers. Users can move freely within the room and even jump to avoid bullets. In Beckhaus' work, the gun aiming is done by the Game Gun, which is also limited to the center of the screen, i.e. the game view will always follow the gun's orientation. A similar game controller concept is used in GameRunner [7]. However, in physical life, the user's viewpoint does not necessarily follow the gun. Our solution of separating the views offers a new approach for FPS games.

Table 1 summarizes the comparison between our system and the others. Our system provides an intuitive and tangible controller in the virtual world and at the same time maintains collaborative/competitive physical interaction between the players. The novelty lies in the ability to aim and shoot other players outside the field of view.

3 System

Our system consists of a game engine and an ultrasonic tracking system (InterSense IS900) with a high accuracy (2-3 mm for position and 0.25°-0.4° for orientation). Its coverage is limited to the space below the sensors, but it can easily be extended by adding more sensors. The tracking system covers a space of $2 \times 4\text{m}^2$ (see Figure 1). The tracking system is connected to a computer (the tracking server), which sends UDP packets containing tracking data (position and orientation) to the tracking clients at a rate of 90 Hz (see Figure 2).

Table 1. Feature comparison of immersive game systems.

System Name	Weapon Aiming Method	Virtual Character's Movement	Accuracy	Ability to Shoot Outside the View Range	Physical Interaction between Players
CAVEQUAKE	Hand	Joystick	High	None	None
CAVEUT	Hand	Joystick	High	None	None
ARQUAKE	Head	Player's Movement	Low	None	None
ChairIO + Gun	Hand	Tilting the chair	High	None	None
Game Runner	Handlebar	Treadmill	High	None	None
Human Pacman	Not Applicable	Player's Movement	Low	Not Applicable	Yes
Touch Space	Hand	Player's Movement	High	None	Yes
Our System	***Hand***	***Player's Movement***	***High***	***Yes***	***Yes***

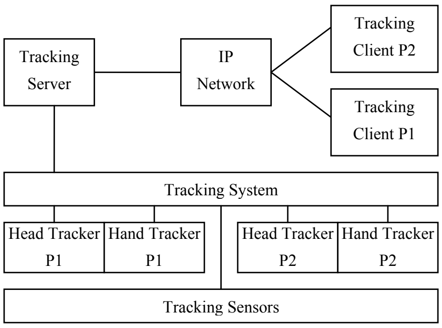

Fig. 2. Tracking system.

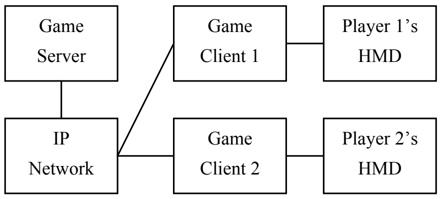

Fig. 3. Game system.

We use the Cube engine [8], an open source game engine written in C++. The Cube engine has a built-in world editor and supports .md2 models. It supports multiple clients through a client-server architecture. In our implementation we use one server and two clients. Each client does the rendering and sends the result into the player's HMD. The server only receives the information of the client's position, orientation, and action. Then the server broadcasts the information to all other clients (see Figure 3).

We implement the system with 4 trackers for 2 players, as each player needs 2 tracking devices (1 head tracker and 1 wand tracker). The head tracker is mounted on the Head Mounted Display (HMD) to track the player's head position and orientation. The game engine renders the virtual environment based on this information. The wand tracker acts as a gun and is used by the player to aim the weapon in game. The player's HMD displays the image as shown in Figure 4. The small window in the bottom right is the gun radar that shows the gun view. With this gun radar, the player can aim and shoot in any direction inside or outside his own field of view. This adds an element of excitement, as the player can aim and shoot the enemy at his back. The white dot in Figure 4 is the weapon pointer. The weapon pointer helps the player aim accurately.

(a) (b)

Fig. 4. In-game screenshots and corresponding player poses.

4 Implementation

4.1 Calibration

The first step to integrate the tracking system and the game engine is calibration. Figure 5(a) shows the real-world coordinate system of the tracking system (where the players move), Figure 5(b) shows the coordinate system of the wand tracker, and Figure 5(c) shows the virtual-world coordinate system of the game engine (where the virtual characters move). The z-axis in the real world points in the opposite direction of the z-axis in the virtual world. The game engine defines yaw as rotation around the z-axis, pitch as rotation around the y-axis, and roll as rotation around the x-axis.

We calibrate the hand tracker to give yaw, pitch, and roll values of 0 when the axes align as in Figure 5(a) and 5(b). Calibrating the position is simple. Because the x-axes and the y-axes of the real and the virtual world are aligned, we only need to adjust the scaling.

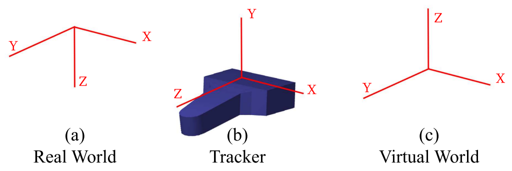

Fig. 5. Coordinate systems.

4.2 Weapon Orientation

Like most game engines, the Cube engine's weapon orientation is pointing to the center of the screen and is tied to the player's view. To make the weapon orientation independent of the player's view, we add new variables for yaw, pitch, and roll of the weapon. We render the weapon's orientation according to the data that we get from the wand tracker. This feature is illustrated in Figure 4. Note that we assume the weapon's position to be the same as the head position.

4.3 Gun Radar

The Gun Radar (a small window that shows the weapon's point of view) is one of the important features of our system (cf. Figure 4). This radar makes it possible to aim and shoot accurately in an arbitrary direction (even if it is out of the player's view). To render this Gun Radar we create a new view port. In this view port we align the player's position and orientation to the gun's position and orientation. Then we render the virtual world in the view port. After that we restore the orientation and the position of the player. Note that the center view of the gun radar becomes the target point.

4.4 Weapon Pointer

The weapon pointer is designed like a laser pointer. It shows the position where the bullet will hit the target. This feature facilitates aiming in the virtual world. It is dependent on the gun radar. We need the z-value of the pixel in the center of the Gun Radar (the target point), which can be read from the z-buffer. Then we have the x, y, and z coordinates of the target point in the frustum. We multiply this point with the inverse projection matrix to get the target point in the game engine's world coordinates. We draw a transparent red dot to mark this area.

4.5 Jumping

If the player jumps in the real world, the virtual character in the virtual world will also jump. To achieve this, we detect the delta-z from the head tracker; if it is greater than a certain threshold, we make the player's avatar jump. This feature can be useful to dodge enemy bullets. It also encourages physical movement. Figure 6 illustrates this feature.

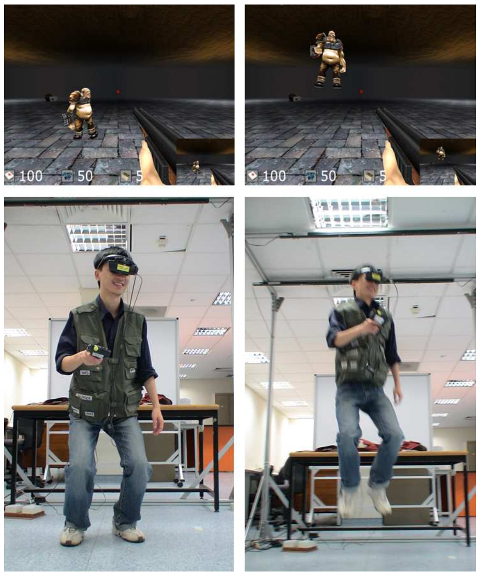

Fig. 6. Jumping.

5 Evaluation

We conducted an informal user study with 30 students (25 males and 5 females). Their age ranged from 19 to 29 years. We conducted 15 sessions of 3 minutes each, in which 2 persons played against each other in a death match battle. For comparison, we also asked them to try the traditional version of the game using keyboard and mouse. After they had tried both interfaces, we asked them to answer the questionnaire. Most of the questions are comparisons between our system and the traditional FPS computer interface. The respondents have to answer on a scale from 1 (prefer traditional implementation) to 7 (prefer immersive system).

We also included some introductory questions to find out about their experience with computers and FPS games. 11 of them had been using computers for more than 10 years, and 23 of them had played FPS games before. The first question was "How would you categorize your self as an FPS gamer?" They have to answer on a scale from 1 (beginner) to 7 (expert). The results are shown in Figure 7. The mean is 3.04. Then we asked how often they play FPS games. The results are shown in Figure 8. Most of them answered "several times" (14 persons).

HMD's are not suitable for long-term use. Playing games with an HMD for more than 15 minutes can cause dizziness in some people. A computer screen is generally better for long continuous game play. We wanted to evaluate how comfortable the HMD was for short-term use (3 minutes). The results on a scale from 1 (computer screen preferred) to 7 (HMD preferred) are shown in Figure 9.

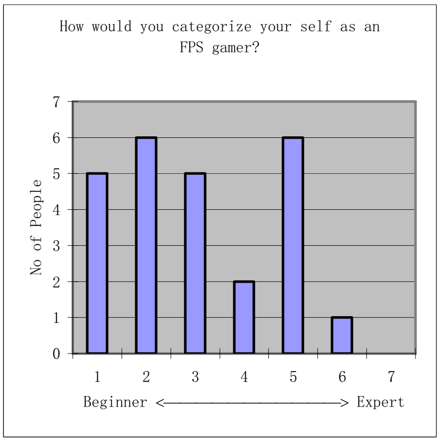

Fig. 7. Skill in playing FPS games.

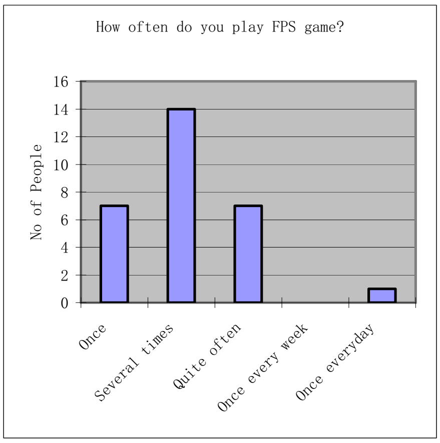

Fig. 8. Frequency of playing FPS games.

The mean is 4.43, indicating that our participants are quite comfortable with the HMD.

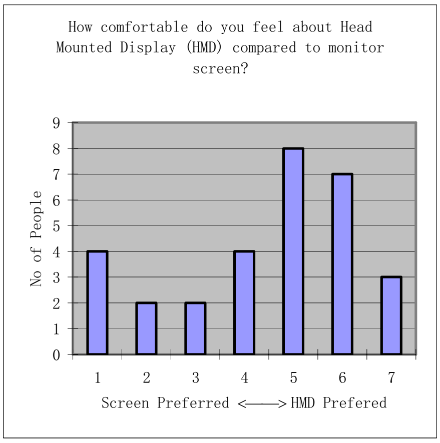

Fig. 9. Comfort of HMD.

We also asked the participants to compare the mouse and the wand in terms of accuracy as well as excitement on a scale from 1 (mouse better) to 7 (wand better). The results are shown in Figure 10. The mean is 3.93 for accuracy and 5.66 for excitement, which indicates that our wand is perceived to be about as accurate as the mouse, but in terms of excitement the wand is preferred.

We also investigate the user acceptance of our new game features (Weapon Pointer and Gun Radar). The results are shown in Figure 11 on a scale from 1 (dislike) to 7 (like). From the chart we can see that most answers are 4 or higher. The mean answer is 5.69 for the Weapon Pointer and 5.17 for Gun Radar. This shows that the participants like our new features.

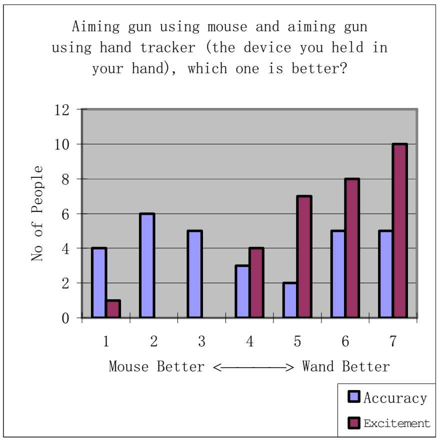

Fig. 10. Mouse vs. hand tracker.

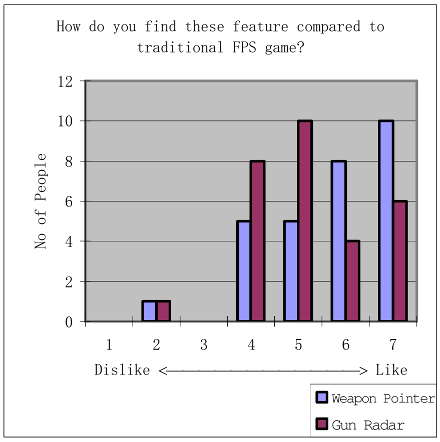

Fig. 11. Weapon Pointer and Gun Radar features.

Our system incorporates two major types of physical movements, namely jumping to dodge bullets, and moving to a particular location to collect certain items (health, armor, bullets, etc). We wanted to know how users feel about these physical movements. Figure 12 shows the results on a scale from 1 (very annoying) to 7 (very exciting). The mean answer is 5.28 for jumping and 4.76 for moving to collect items. This shows that physical movements are well liked by the users. Jumping gives more excitement than just moving around by walking.

In the last question we asked “Overall, How do you feel about using physical body movement to play FPS compared to using traditional keyboard and mouse?” The results are shown in Figure 13 on a scale from 1 (prefer traditional interface) to 7 (prefer our system). The mean answer for this question is 5.59, which shows that our system is significantly more attractive than the traditional keyboard and mouse interface for FPS gaming.

6 Conclusions and Future Work

We have built an immersive system for FPS games using an ultrasonic tracking system. The system encourages tangible and physical interactions between players, especially in a tense competitive situation. It allows users to move about and look around freely while aiming at targets using a gun-like handheld device. We use a novel way to present the user’s view and the gun’s view separately. Jumping is also tracked so that the user has more options to avoid the bullets.

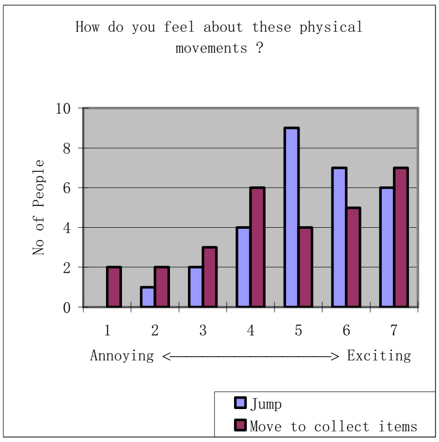

Fig. 12. Physical movement.

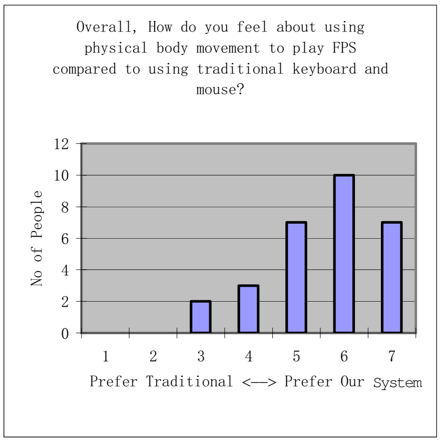

Fig. 13. Comparison of interactions.

Our user study shows that people enjoy our system more than the traditional computer-based FPS interface on many levels.

References

1. Beckhaus, S., Blom, K.J., Haringer, M.: A new gaming device and interaction method for a first-person-shooter. In: Proc. Computer Science and Magic, Leipzig, Germany (2005)
2. Rajlich, P.: CAVE Quake II. `http://brighton.ncsa.uiuc.edu/~prajlich/caveQuake/`
3. Jacobson, J., Lewis, M.: Game engine virtual reality with CaveUT. Computer **38**(4) (2005) 79–82
4. Thomas, B., Close, B., Donoghue, J., Squires, J., Bondi, P.D., Piekarski, W.: First person indoor/outdoor augmented reality application: ARQuake. Personal Ubiquitous Computing **6** (2002)
5. Cheok, A.D., Fong, S.W., Goh, K.H., Yang, X., Liu, W., Farzbiz, F.: Human Pacman: A sensing-based mobile entertainment system with ubiquitous computing and tangible interaction. In: Proc. NetGames (Workshop on Network and System Support for Games), Redwood City, CA (2003) 106–117
6. Cheok, A.D., Yang, X., Zhou, Z.Y., Billinghurst, M., Kato, H.: Touch-space: Mixed reality game space based on ubiquitous, tangible, and social computing. Personal Ubiquitous Computing **6**(5-6) (2002) 430–442
7. GameRunner. `http://www.gamerunner.us/`
8. Cube Engine. `http://www.cubeengine.com/`