The Invoker: Intuitive Gesture Mechanics for Motion-based Shooter RPG

Albert Quek  
Faculty of Computing and Informatics  
Multimedia University  
Cyberjaya, Selangor, Malaysia  
Email: quek.albert@mmu.edu.my

John See  
Faculty of Computing and Informatics  
Multimedia University  
Cyberjaya, Selangor, Malaysia  
Email: johnsee@mmu.edu.my

Abstract—The introduction of motion controller and camera as gaming peripherals have increased the immersion of computer games today. Moving on from using conventional handheld game controllers, players can now use gestures for gameplay control. At this juncture, despite the popularity of Shooter Role Playing Games (SRPG) among developers and researchers, gesture mechanics remains limited and underdeveloped. In this paper, we have experimented with a set of simple intuitive gestures for a newly designed SRPG called The Invoker. Implemented with the Microsoft Kinect, The Invoker emphasizes on the battle experience by utilizing hand and body movements to dodge and cast spells. A selection of gesture transitions were designed to produce various skill combinations during gameplay. For further enhancement and usability, visual and audio feedback were also provided to the player when an action is performed. By qualitative user evaluation on a target RPG interest group, about 70% of the participants expressed positive feedback on the design of spell casting gestures and skill combination mechanics in the game.

Keywords—Gesture-based game; Gesture mechanics; Intuitive gestures; Microsoft Kinect; Shooter RPG.

1. INTRODUCTION

As games become more sophisticated in terms of visuals, gameplay complexity, interaction and narrative, it have branch out into many genres that dominates the market. This is defined in the Herz System where the major categories of games are action, adventure, fighting, puzzle, role-playing, simulations, sports, and strategy game [1]. A role-playing game (RPG) is a game in which player acts as the role of a character, experiences the fantastical adventures, basically in fantasy or fiction setting. Literally, RPG requires players to make decision and carry out the action within a narrative system as they are interacting in games imaginary world. Electronic RPGs are often consisted of narrative story, combat system, character growth system, missions and quests [2]. In creating a believable character in RPG, it requires several qualities to make it as believable: human body; self-awareness, intentional states, and self-impelled actions; expression of emotions; ability to use natural language; and persistent traits. For example, a creature is believed to be a human if he/she is having a human body form and able to communicate by using natural language as we are. Besides that, the statement indirectly implied that human will easily believe to the traits that similar to them [3]. Therefore, in this paper we have explored the use of First Person view and intuitive gestures in creating a believable playable character. Brathwaite and Schreiber [4] view character development as the most important part of any RPG as it is another method to make the character to be believable.

Battle system design is also important, as players spent a large part of the gameplay time in battles and preparation for battles are often the second most time consuming gameplay part [5]. Designing of the battle system can be categorised into Western RPGs, Japanese RPGs, Action RPGs, Tactical RPGs, Multiplayer Online RPG and Shooter RPGs (SRPG) [5], [6]. We have identified SRPG as our battle system design as they are a hybrid between that genre and First person Shooters. We also explored the possibilities of using gestures in designing the battle system for SRPG. Similar SRPG such as Fable: The Journey and Harry Potter Kinect uses gesture control for the gameplay mechanics. Fable: The Journey uses both hands to cast spell and perform aiming of enemies. No aiming cursor shown in the game to increase the realism of the battle. Player has to perform push gesture to trigger the attack [7]. One main flaw reported in its Metacritic review [8] relates to the unresponsiveness of controls while overall gameplay was also found to be repetitive at times. On the other hand, Harry Potter Kinect has a simpler gameplay which includes a collection of casual mini-games [9]. Due the simplicity of its gameplay, spell casting and voice commands are effectively incorporated with the use of Kinect. The right hand is used for targeting enemies in battle while a push motion activates spell casting. At Brown University, an experimental gameplay named Swordplay was developed in an attempt to emphasize player control and interaction using natural human movements rather than conventional control mechanisms [10]. Using the Nintendo Wii Remote controller, the player is able to perform simple gestures such as swinging the sword, firing an arrow, drawing spell symbols or dodging attacks. For instance, the sword swinging can be easily learnt by mimicking the motion of drawing a weapon. Swordplay still somewhat relies on some artificial controls from the Wii Remote, such as pressing a button to start a spell and navigation with an analog stick; hence, not entirely gesture-based.

Kristensson et al. [11] proposed a gesture-based interface that supports both one-handed and two-handed gestures to directly issue commands. They extended their previous work of a continuous recognition algorithm for the Kinect to accommodate simultaneous gestures from both hands. A high accuracy rate of 92–96% was achieved on their collected data set. This work provides further motivation to utilize gestures from both hands to invoke game actions in a markerless sensor.
In this paper, we proposed a new motion-based SRPG called The Invoker that utilizes a set of simple hand gestures and body movements that are both intuitive and gameplay enhancing. During gameplay, these gestures can be combined to yield a variety of offensive and defensive skills. Visual and audio feedback were also provided to heighten user experience during skill invocation. The rest of the paper are organized as follows: Section II introduces the game concept behind The Invoker and the proposed user interface design. Section III describes the gesture set and its entire mechanics implemented in the proposed system. Finally, Section IV presents some user evaluation results while Section V concludes the paper.

II. GAME DESIGN

A. Game Concept

The Invoker is an interactive first-person gesture-based Shooter Role Playing Game (SRPG) implemented with the Microsoft Kinect. The player takes up the role of a wizard apprentice protagonist who has to overcome a set of level puzzles, challenges and boss battles in order to become a master wizard. The game is fantasy-themed; with four level settings depicting violet sky, cave, mountain and misty forest scenes. In each level, players are required to learn unique spell casting gestures in order to clear puzzles, and fight monsters. The left hand is used for defensive gestures while the right hand is designated for attacking gestures. Throughout the levels, the player has to master five different spells: Fireball (2 types), Waterball (2 types), Healing. In the first three levels, the player will be tested on the mastery of different aspects of spell casting. For example, Level 1 tests the player’s accuracy in aiming; Level 2 tests the player’s agility; Level 3 tests the player’s puzzle solving skill while level 4, being the last level, is a combination of all.

B. User Interface

In the game, we have designed our interface in a simple yet informative way to prevent cognitive overloading [10]. The demands of having to use gestures on both hands need to be well balanced with a straightforward user interface that is easy to understand. Basically, the interface that we have designed provides relevant in-game information to the player so that player is able to make quick and good decisions when battling with an enemy. For instance, the horizontal bar and the skill panel are clearly displayed to aid the player to decide between perform a regeneration of the players current health points (HP) or to continue casting an attack spell onto the enemy (see Figure 1).

1) Player’s Stats: The health points (HP) and mana points (MP) of the player are shown by the horizontal bars at the top left corner of the screen. Player is unable to cast spell if the MP reaches 0. Player will lose the game if the HP is reduced to 0.

2) Skill Panel: List of skills that are ready for casting is shown in this panel. There are only three slots

3) Enemy’s Stats: The health points (HP) of the enemy is shown by the horizontal bar at the top right corner of the screen. The enemy is killed if its HP is reduced to 0.

C. Implementation Tools

The Invoker was developed using Microsoft Visual C# with Microsoft Kinect SDK [12], supplemented by various programming interface (APIs) that are supported by Microsoft Kinect drivers to control the audio, video data streams and skeletal tracking of the player [13]. We also made use of the popular Digital Rune Graphics Engine [14] which encourages rapid development of 3D worlds with cutting-edge collision detection and particle systems.

Generally, our implementation enables the detection of gestures for both right and left hands, thus allowing players to use both separately, or simultaneously for complex spell casting that requires combinations. The animation captured via Kinect sensor for each hand is mapped using the skeleton mapper from Kinect SDK to allow the translation of the players physical hand movements into game instructions and on-screen graphical display of the avatars hand.

III. GAME INTERACTIONS

The Invoker is a gesture-based SRPG that relies heavily on hand and body movements to perform various actions. We have designed both attack-oriented and defensive-oriented gestures such as spell casting, dodge, shield, and regeneration. Spells such as Fireball, Waterball and Heal can be cast using a combination of different geometrical shapes drawn during the spell casting mode. A tutorial level was created to allow the player to familiarize with the different gestures. During the game, visual (avatar hand movements) and audio (special effects) were provided to the player when the player performs an action. Players are required to memorize these gestures and/or combination of gestures that would produce the intended action.

Next, we describe the set of basic and special gestures designed for player actions in the game (Section A), followed by an elaboration of the gesture mechanics involved in the game (Section B).

A. Gesture Set

We construct the basic gesture set by designing a collection of different gestures that were adapted from recent Kinect-based games [9], [10] that were relatively successful. The gestures are described as follows:
I) Basic Gestures:

- **Ready**: Standby gesture to prepare players avatar for casting a spell. Player has to swipe up his/her right hand to be ready for casting spell.

- **Dodge**: Player can dodge attack from enemy by moving the body to the left or to the right. The screen will move to the left or right depending on the players movements.

- **Spell**: In Ready mode, player can cast a spell using the right hand to draw three possible shapes: Circle, Triangle or Inverse Triangle, which determines the type of spell to cast. When the spell is ready for casting, a symbol will appear on the skill panel. Different combination of shapes can be used to create different spells. Player is allowed to combine up to a maximum of 3 types of shapes (by drawing continuously) to cast different types of spells.

- **Cast**: Player can aim at the enemy and hit them with the casted spell by pushing the right hand in a forward motion.

- **Regeneration**: Player can regenerate the characters mana by moving the left hand 45 degrees from the waist, holding it still for 5 seconds.

- **Shield**: Player can generate a shield to block an enemy attack by lifting up the left hand.

The use of the right and left hands provides an intuitive separation between attack-oriented and defensive-oriented gestures. For instance, it is easy to cast spells with the right hand, freeing up the left hand to shield from quick enemy counter-attacks. Also, Regeneration is intuitively regarded as a defensive-oriented gesture, hence the use of left hand as well. The 5-second hold-still rule is in place for game balancing purpose; to prevent over-usage of this action by the player. Dodging, albeit defensive in nature, is best captured by body movements rather than hand gestures. See Figure 2 for a pictorial illustration of some of these gestures.

2) Special Gestures:

- **Power-Up**: The player can power-up by collecting mana points quickly by using both hands. The player will be instructed on how the gesture is to be executed by both hands. This is enabled only in the last level.

- **Multi-Attack**: The player can also mount multiple attacks in quick succession by moving forward both hands. This is also enabled only in the last level.

B. Gesture Mechanics

We have experimented with two methods of recognizing gestures: linear gestures and template gestures. Linear gesture method is easy to implement, and it is also most useful and accurate when the desired movement is linear either from left to right, up to down and vice versa. Hence, it can successfully detect simple stationary points, lines or slashes. However, this method is less accurate for detecting geometrical shapes such as circles, squares and triangles. Therefore, the template gestures method is most suited in this case to a good measure of success [11]. Nevertheless, it still struggles with the difficulty of distinguishing between shapes that are of very similar nature (e.g. circle and oval, square and rectangle).

In our game, we employ both linear and template gesture methods to cater to the proposed gesture set. To detect linear gestures, the position, length and duration of the hand movements were identified as the features. Meanwhile, for the detection of template gestures, a database containing a number of pre-defined geometrical shapes that are to be detected in the game is first created. The system will then recognize the pattern if they are drawn within the defined range of tolerance around the relative position of coordinates of each shape. Figure 3 summarizes this concisely in pseudocode. A player will perform a gesture, which will be compared with the existing gesture templates stored in the database, within a fixed time limit. The detected gesture is then identified and passed to the system to update the necessary gameplay elements on-screen. In our game, we used three different shapes that are straightforward and can be drawn sufficiently quick: Circle, Triangle and Inverse Triangle (see Figure 2 for the shapes).

The linear gestures Swipe Up and Swipe Down are used to turn on and turn off (respectively) the Spell or Shield commands. If movement in the vertical dimension (Y) is more than the pre-set gesture length within the time limit, it will return the appropriate swipe command, depending on the direction of the gesture (upwards or downwards). Meanwhile, the Push command, which involves a horizontal linear gesture along the depth dimension (Z), is only used to initiate the Cast command. Figure 4 summarizes the logic behind these linear gestures.

The attack- and defensive-oriented gestures are mainly controlled by the right and left hand respectively. When the player moves the right hand upwards (Swipe Up), Spell action is activated and the system is ready for spell casting. The system then detects up to three shapes/gestures that are drawn...
by the player, terminating with the Push gesture. When players
hand pushes outwards to the front after completing the shape
combinations, the system will trigger the Cast action to cast
the desired spell towards where the player has aimed at. Figure
5 shows a summary of the right hand attack gestures.

The type of spell selected is determined by the combination
of two different shape gestures drawn by the player. This is
selection process is concisely represented by the skill decision
tree in Figure 7. There are altogether 5 different possible
spells that the player can cast, namely Fireball High, Fireball
Low, Waterball High, Waterball Low and Heal. Upon casting
the desired spell, the players mana points are also deducted
appropriately. The left hand gestures represent defensive-
oriented actions. When the left hand swipes upward, the system
will activate the Shield command for the player. This also
comes with a time-scaled consumption of mana points. More
importantly, Shield can be used along with other right hand
attack gestures and this is crucially important during battles. At
times where the player requires a fast recovery after battle, the
player can move the left hand 45 degrees from the shoulder,
activating the Regeneration command. This replenishes the
players mana points at a faster rate, but disallows all other
attacking actions. Figure 6 shows a summary of the left hand
defensive gestures. Figure 8 shows a selection of screenshots
depicting the various gestures designed for The Invoker game.

IV. USER EVALUATION

The Invoker game was created to introduce intuitive gesture
mechanics into a Kinect-based fantasy-themed SRPG. Hence,
the target audience were primarily people who have a strong
interest in fantasy role-playing games, and have played them
often. The game was tested with 13 participants ranging from
18 to 20 years old. All participants were given only 15
minutes to play with an option to extend playing time. Through
our qualitative survey and interviews, we found that 70% of
the participants expressed a strong liking towards our design
and implementation of the basic action gestures and skill
combination gestures in the game. On average, the participants
requested an additional 10 minutes of time extension as they
felt immersed with the action in game. All gestures and game
mechanics were thoroughly tested and we observed that all
participants were able to complete the game with minimal or
no assistance.

The game was also showcased in the Science Festival
received were mostly positive as walk-in participants were able
to perform both attack-oriented and defensive oriented gestures
correctly. The participants were satisfied with the gameplay
experience, through the introduction of gestures mechanic in
the game.
V. CONCLUSION

In conclusion, we have proposed a new shooter role-playing game (SRPG) using intuitive gesture mechanics. Both linear and template gesture methods were designed to cater to various proposed gestures; the former for quick and simplistic actions, the latter for more complex gesture combinations. We also proposed an intuitive separation of roles assigned to each hand (attack-oriented gestures for right hand and defensive-oriented gestures for left hand) to enable execution with minimal confusion. Spell casting skills in particular, were devised through the use of various shape-based gestures. To further enhance gaming experience, audio and visual feedback were also provided to the player during game. In a qualitative evaluation on a target RPG interest group, about 70% of participants responded positively towards the design of action gestures and spell casting gesture combinations in the game.

In the future, we will look into incorporating other possible gestures that may appear more natural and instinctive in shooter role-playing games. This may include recognizing multiple simultaneous movements from both hands, and shoulders.

ACKNOWLEDGMENT

The authors would like to thank Tan Zhi Wei and Aaron Chen Wei Ming for developing the game project prototype, Multimedia University, Telekom Malaysia and Petrosains for providing support and the necessary testing environment.

REFERENCES