Game Mechanics: Sniper Simulator Game

A Development Study for Animations and Interactivity Game Prototype

Elbert Christopher Larosa
Computer Science Program
Bina Nusantara University International
Jakarta, Indonesia

Matthew Kharli
Computer Science Program
Bina Nusantara University International
Jakarta, Indonesia

***Abstract*— Popularity of game creates opportunity to everyone in developing a game that can be used to address any specific problem in many areas. In BINUS University International, for education purposes, game is not only being used as a tool/media to learn something, but it can also be used to develop student understanding on how the game mechanic of the games works. By understanding the mechanic of the games, hopefully it can help them in developing a game designed that can be used to address problems in their environment.**

Keywords—Educational Game, Game Designed, Game Mechanic

I. Introduction

Currently, game becomes more popular in Indonesia. Supported by the advanced of current technologies, game engages many people by creating an environment that can provides entertainment and enjoyment. Besides, game can also be utilized to many areas, which make game are widely used today. For education purposes, a number of institutions have been adopted and widely used a game to educate and promote good behavior in their workplace [1]. For school and university in Indonesia, some of them have been designed a curriculum that related with game areas. A number of subjects at school teach about game in the classroom to trigger students on how to understand about the game design and mechanic[2][3].

Based on the book by Rouse [4], game designed can be defined as something that determines every detail of how the gameplay will function. For example, it determines the play form of the game, the choices that allows players to be able to make in the game-world, and the ramifications of those choices that have on the rest of the game. The game design can also determine what win or loss criteria the game may include, how the user will be able to control the game, and what information the game will communicate to the player, and it establishes how hard the game will be.

In developing a game for some specific purpose, it is important to learn how a game is designed. It is also important to understand the game mechanics thus it can be used to address the objective of the game itself. Game mechanics can be defined as series of rules or method that is required to make interaction between the player and the games stage, hence providing a gameplay. All games have their own mechanism that makes each individual game unique. Hence, game mechanic plays important role on how game is designed[5][6].

In addressing the potential of game development in Indonesia, BINUS University International is embedding game streamline as part of their curriculum. With this streamline, the students are expecting could learn not only on how to play a game, but also learn on how to develop a game designed that can be used for any purposes in many areas. Hence, in this paper, we tried to explore and describe the game designed and game mechanic of a game prototype, which can provide more information and understanding to other people that might be interested in developing a game.

This paper will discuss more detail about the design and game mechanic of a digital game that we call “Sniper Simulation Game”. This paper tries to focus on digital game as one of game types that popular in Indonesia society. It also describes the scenario of gameplay and the benefit and limitation of this game based on the design and mechanical perspective. A digital game prototype will be developed as the product of the game designed and game mechanics.

II. Background and Application’s Description

As already mentioned above, the end result of this development study is creating a digital game prototype that could provide more information on the game designed and game mechanics. Experiencing prior knowledge in the animations and interactivity user interface. In addition to have interest in the subject of sniping (shooting at individuals as opportunity offers from a concealed or distant position), and also because of having early understandings in some of sniping’s concept, we decided to make a realistic Sniper Simulator game.

The game, titled “Sniper Simulator Game” is a game where the player will act as a sniper recruit that is currently being tested by the academy, and that the recruit has to hit all the target practice placed at various positions and ranges. The player’s number of shots is limited and that the player must hit all the targets with the available bullets in order to pass. Although seemed like there is no obstacle, the player must fight against wind and gravity that could affect the shot, resulting in a miss.

In this game, multiple animations are added such as the target falling down when hit by the bullet. There is also a user interface that contains helpful information for the player to calculate the bullet movement after being affected by the wind and gravity (such as the distance info and wind meter). The wind and gravity effects are calculated using formula made especially for the game to enable the mimicking of those effects. Multiple interactivities are also added such as viewing, shooting, and zooming controls. The detailed description of the game's features is going to be discussed in the Project's Features and Interactivities section.

III. MODELING TECHNIQUE DESCRIPTION

For the game's modeling technique, the models in general are divided by 2 parts; the terrain – rifle and the target model (using imported 3D models and then processed using Blender software). The rifle and target models are resized to fit the terrain size ratio, so that the size of the rifle and the target would not be too big compared to the terrain. The rifle model is then placed in a fixed position on the terrain, and that in the end the rifle and the terrain models are merged together. While the target model is a stand-alone model, and that in the program 10 different target positions are pre-defined first. Then, when the game is initialized, 5 of the positions are chosen randomly for the 5 targets (thus when the game is reloaded, the target positions would be different because it is randomly set).

The models have been textured using images, such as the target model using 2 different images to texture the target's rope and body, and the terrain model using 3 different images to produce varied texture looks for the terrain. In the game, the shadow effects also work, which is by using 2 lighting sources. One of the lighting sources is put on top of the scene and is heading toward the scene in a perpendicular manner (thus giving an overall lighting for the scene). While the other lighting source is put from a diagonal angle, thus producing a shadow effect for the models in the game. Lastly, the sky model is produced with the help of SkyShader.js, which enables the production and configuration of the sky's characteristics, such as turbidity and luminance.

IV. PROJECT'S FRAMEWORK

The game is built and tested using a computer with the specification listed below:

- Windows 7 Home Premium 64-bit
- 8192MB RAM
- Intel Core i5-4200U CPU @ 2.80 GHz (4 CPUs)
- NVIDIA GeForce GTX 850M 2048MB GDDR5

The game is tested on the browsers Google Chrome version 50.0.2661.102 m. and Firefox version 46.0.1.

The project is using WebGL as the renderer for the game's visuals, which are 3D based. For the framework, the project is using Three.js along with its implementations, such as:

- FirstPersonControls.js, which is used for first person view controls. Other controls such as mouse click, and mouse scroll along with their effects are defined manually.
- OrbitControls.js, which is used for debugging and early game development because the Orbit Controls give a more flexible controls for the camera to move around the whole map.
- MTLLoader.js and OBJLoader.js, which are used for importing the object and material files for the game's models (such as the terrain, sniper rifle, and the target) to the browser.
- SkyShader.js, which is used for creating the sky for the game's environment.
- tween.min,js, which is used for creating animation movements for the game, such as:
  - Camera movement of the player approaching the rifle scope (in the beginning of the game)
  - Target down animation when hit
- Clock.js, which is used in this game to enable updating based on time (such as keep updating every second).

V. PROJECT'S FEATURES AND INTERACTIVITIES

The project's features and interactivities are going to be explained by dividing into some sectors; which are the game's animation, controls/interactivities, bullet's movement calculations, user interface, and audio used. The animations made in the game are:

1. In the beginning of the game, there will be a camera movement animation where the camera will move to approach the riflescope. Then, the view will fade to black and then the camera view will change to the riflescope view, where the player then will be able to do the controls.
2. The target down animation, which triggers when the target is successfully hit by the bullet (bullet's movement calculation is going to be explained in the bullet's movement calculations sector). In this animation, the target will fall down and then disappear from the scene when it is hit.

The controls and interactivities made in the game are:

1. First person view controls, which is by moving the mouse towards the direction the player desire. The camera will be able to look through certain angles, but it will not be able to move (always stay in the same position).
2. Shooting controls, which is by clicking the mouse. When triggered, the shot audio will be played and a recoil animation will be triggered, which is the camera being forced to see upwards for a short while to mimic a recoil effect.
3. Zooming controls, which is by using the mouse scroll. The riflescope can be zoomed in and out to a certain zoom level.

Bullet calculation in this game is using Ray caster to help determine whether the shot hit the target and to help find the distance between the rifles to the target hit. Here are the explanations regarding the bullet's movement calculations:

1. Gravity

The gravity effect will cause the bullet to drop down as it travels further distances in a parabolic matter (affecting the y-axis of the shot). In this game, a quadratic formula is made to mimic the gravity effect and to fit with the available targets' distances:

Gravity Effect = (0.002282 x distance2) – (0.1098 x distance) + 0.4584

With the formula, it is ascertained that the furthest target will get the most significant gravity effect, without getting to far from the scope. The resulting number from the calculation can then be used to measure how much higher should the player aim in order to compensate the bullet drop effect. There is the gravity effect value marker on the scope to help the player compensate the gravity effect. The way distance value is retrieved will be explained in the user interface sector.

2. Wind

The wind effect will cause the bullet to sway left or right (affecting the x-axis of the shot), depending on where the wind is currently heading. The direction and intensity of the wind are randomly generated (the values are still limited so that the bullet will not sway too far left or right), but there is a dynamic wind meter UI provided to help the player compensate for the wind effect.

The user interfaces made in this game are:

1. Bullets Left UI, showing the number of bullets remaining before the player run out of any more chances. The number of bullets will be decreased as a shot is done.
2. Target Remaining UI, showing the number of target(s) remaining. The number will be decreased when the user successfully hit a target.
3. Wind Meter UI, showing the current wind's direction (left or right) and intensity. The player must adjust the aim to match the current wind meter's pointer, in order to compensate the wind effect.
4. Distance UI, showing the distance from the rifle to a target. The distance UI will show a distance value when the crosshair is hovered toward a certain target. The player can then calculate the gravity effect from the given distance value.
5. Winning UI, showing a winning condition statement and stopping the game controls. The player wins when the player is able to hit all targets with the available number of bullets.
6. Losing UI, showing a losing condition statement and stopping the game controls. The player loses when the number of bullets available has run out but there are still target(s) remaining.

The audios used in this game are:

1. Target Hit audio, which is triggered when a target is hit.
2. In-game BGM, which is triggered from the beginning of the game and is looped until the player wins / loses.
3. Starting audio, which is a vocal sound recorded to indicate the start of the game.
4. Winning BGM, which is triggered when the winning condition is achieved.
5. Losing BGM, which is triggered when the losing condition is achieved.
6. Rifle Shot audio, which is triggered when the mouse is clicked (the player shoots).

VI. Application Manual

Below is the application manual that explains how the game is played:

Firstly, the player will encounter an animation of the player approaching the riflescope. Then the view will be changed into the rifle scope and the player is allowed to do the controls. The player can take a look of the surroundings and find out the 5 targets placed randomly in the environment. The player can also zoom in and out to get a better view of the target.

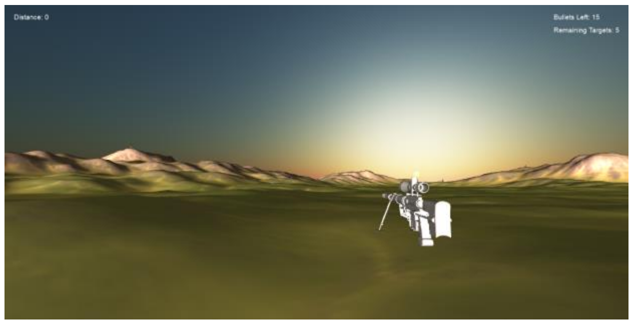

Fig. 1. An animation in the beginning of the game of the player approaching the riflescope.

Secondly, to hit a target, the player must firstly hover over the target first to find the distance value. Then, the player can calculate the gravity effect based on the distance value showed by the UI, which is by:

i. Dividing the distance value showed from the UI by 100
ii. Use the value to the quadratic formula specified earlier, which is: Gravity Effect = (0.002282 x distance2) – (0.1098 x distance) + 0.4584

Adjusting the scope to aim higher can use the value retrieved from the calculation, fitting the value of the scope's gravity value marker to the gravity effect value calculated earlier.

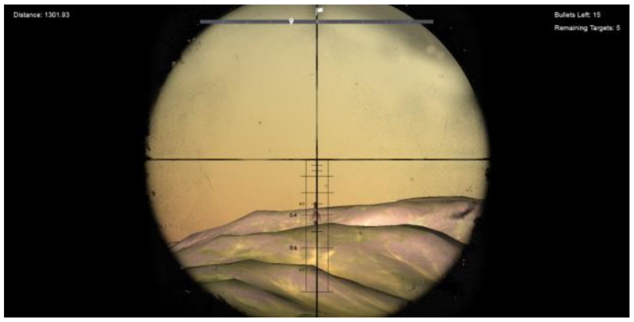

Fig. 2. For the specified distance, the gravity effect would be approximately 0.4. Thus we compensate by aiming higher, adjusting the aim to the 0.4 marker in the riflescope.

Thirdly, the player must also consider the wind effect, which is by taking a look to the wind meter's pointer and adjusting the position of the target in the rifle's aim to the current wind meter's pointer position.

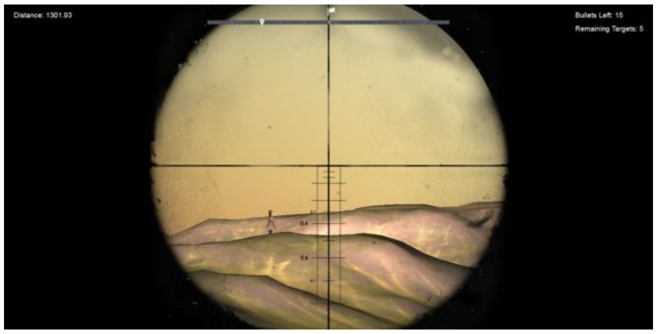

Fig. 3. After compensating for gravity effect, we consider the wind effect; which is by adjusting the aim to match the pointer position of the wind meter.

Fourthly, after the player has adjusted the rifle aim to fit the gravity and wind effect, the player can click the mouse to shoot.

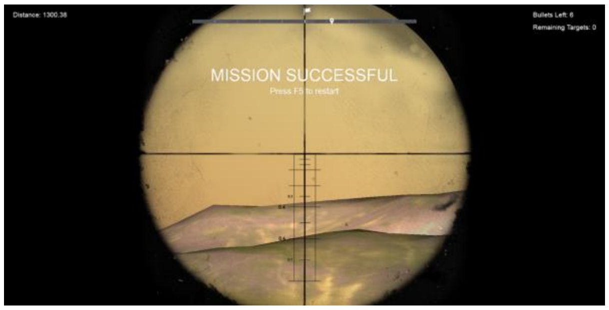

Fig. 4. The winning condition UI, which is triggered when all the targets have been successfully hit.

Lastly, do the same steps (step 2-4) and hit all the targets with the given number of bullets in order to win the game. In the end of the game (winning or losing), the player can press F5 to restart the game.

VII. CONCLUSIONS

In this paper, we have described and introduced an information regarding to the development of game mechanic for Sniper Simulation Game. Here, a game with a concept of sniper simulation that simulate player to act as a sniper recruit that is currently being tested by the academy.

In the game mechanic we have discussed how to set the player's number of shots where is limited must hit all the targets with the available bullets in order to pass. In addition, we also have created a formula that situated seemed like there is no obstacle, but the player must fight against wind and gravity that could affect the shot, resulting in a miss.

This game prototype is still an early stage of development, there are a lot of improvement still can be added to increase the excitement of the player to play the game. For further work, it is important to test it this game mechanic to see how well the game mechanic could be implemented into a game application.

ACKNOWLEDGMENT

This work is part of a class project for Computer Graphics in Bina Nusantara University International.

REFERENCES

[1] Y. G.S, "Mari Belajar Nilai-Nilai Anti-Korupsi Sambil Memainkan Board Game," *boardgame.id*, 2018. [Online]. Available: https://boardgame.id/main-belajar-anti-korupsi-board-game/. [Accessed: 16-Sep-2019].

[2] J.-N. Proulx, M. Romero, and S. Arnab, "Learning Mechanics and Game Mechanics Under the Perspective of Self-Determination Theory to Foster Motivation in Digital Game Based Learning," *Simul. Gaming*, vol. 48, no. 1, pp. 81–97, Feb. 2017.

[3] S. DuBravac, "Game Mechanics for Classroom Engagement," 2012, pp. 67–94.

[4] R. Rouse, *Game design : theory &amp; practice*. Wordware Pub, 2005.

[5] Boardgamegeek, "Game Mechanic," vol. 2009, no. 16 November. 2009.

[6] I. B. K. Manuaba, "The Design and Game Mechanic of Combined Game Application Prototype for Learning Social Business," *Procedia Comput. Sci.*, vol. 135, pp. 52–59, 2018.