

PAPER • OPEN ACCESS

Historical Theme Game Using Finite State Machine for Actor Behaviour

To cite this article: M F Syahputra *et al* 2019 *J. Phys.: Conf. Ser.* **1235** 012122

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the **collection** - download the first chapter of every title for free.

Historical Theme Game Using Finite State Machine for Actor Behaviour

M F Syahputra¹, A Arippa¹, R F Rahmat¹, U Andayani¹

¹Department of Information Technology, Faculty of Computer Science and Information Technology, University of Sumatera Utara, Medan, Indonesia

nca.fadly@usu.ac.id | andean_arippa@students.usu.ac.id | romi.fadillah@usu.ac.id | ulfi.andayani@usu.ac.id

Abstract. North Sumatra has a lot of stories about the history of the struggle. For instancing on the Dutch military aggression in Indonesia after independence. Many insurgencies do each region in northern Sumatra to maintain its independence. This application was created to show the game in the form of 3D for entertainment and education. This game genre of First-Person Shooter in which players seek to kill the enemy and shot her way. This game application design based on the folklore of North Sumatra on fighting Stabat from Dutch hands. Game Engine used is Unity 5. Unity can be used to produce the game in 3D to realize the real world in virtual form. The design of this game is the result of the analysis, design, and implementation of methods Finite State Machine is used for NPC behaviour. Results from the design of this game is a game in 3D based on the story of history and to NPC using Finite State Machine.

1. Introduction

Technological developments are happening at this time has made a significant change in the pattern of human life. The need for technology becomes almost evenly distributed throughout the world. Most people use this technology to support their life and as a means of entertainment, one of the uses of computer technology as a means of entertainment is a game. Game in great demand because it requires interaction with the user. Ranging from children to adults like Game. In general, the game is made for entertainment for its users. But that did not make the game 100% safe for consumption. The things that make the game it is not safe to eat is the rating of the game, the visual effect output Game, cultural differences between the user and vendor Game, and others [2].

Video Game is a "game" with a picture or a visual display that can provide a response back if given certain commands using controller on a set of electronic system [3]. Each game type has two properties, namely 2D and 3D visualization. Game at this time a more interesting game play is based 3D because the object contained in the game look more real [4]. To create a 3D-based games needed.

Game Engine to make it happen. One of the famous Game Engine is Unity. At this time a lot of game genres, one of which is a First-Person Shooter (FPS). FPS Video Game is a genre that focuses on weapons and weapons of war through the projectile-based first-person perspective, which the player experience through the eyes of his protagonist [5].

One component that is now applied to the game is the artificial intelligence (Artificial Intelligence, or AI). AI is defined as the Scientific Intelligence Entity. Research on AI (Artificial Intelligence) at the NPC (Non-Player Character) in a game until now continue to be developed. Artificial Intelligence was developed for designing the behaviour of the Non-Player Character. Artificial Intelligence in Game First



Person Shooter generally consist of planning the path, picking up items, use items, and fight. Especially for fighting Non-Player Character is also expected to have specific strategies such as humans. The strategy is meant to be a strategy where the decision when to pursue the opponent, attacking the opponent or opponents with adjustments to avoid Non-Player Character condition at that time [6]. One method used to make the artificial intelligence in the game is the method of Finite State Machine (FSM). FSM is widely used to make the game in certain situations can produce a response. In Game, story is a very important part.

This study uses an algorithm that is commonly used in most major games such as Counter-Strike [9]. In 2011, Mizanuddin make First Person Shooter games Operation Lun Inau using Unity. Games created as a means of entertainment is quite interesting [2]. Meisadri et al (2013) examined the development of 3D First Person Shooter game Alien Hunter using the A * Star as path finding and fuzzy logic [10]. This game is adapted based on the book THE STRUGGLE OF THE NORTH SUMATRA published by the foundation preservation of fact struggle in 2007. Previous studies conducted by Peng Xia 3D game development with Unity. Casing study A First-Person Shooter (FPS) game.

2. Concept Development

2.1. Data Collection

Data collected in the form of photos train station as an initial start in the game. In Figure 2 can be seen some of the photos that have been collected which forms the basis and reference in making 3D buildings in Stabat. That photos many taken from Dutch museum that has long official online, such as the *Koninklijk Instituut voor Taal-, Land- en Volkenkunde* (KITLV) managed by *Leiden University* in the Netherlands and *the Tropenmuseum* (Museum of the Tropics).

In addition to photograph the station, which is no less important is the Stabat map photos at the time. Dutch East Indies colonial government had mapped the areas of East Sumatra, including the *Stabat* and *Tanjung Pura*. As in Figure 3 that is needed to make the terrain.

2.2. Modelling

With the limited information available, in determining the size of the house, investigators calculated based on the scale and the estimation of the actual size of an object photographed in the train station. The method used in this 3D modelling process is *image-based modelling* (IBM) using *Sketch Up* 3D software. For example, as shown in Figure 4. The 3D characters in these applications created using *Adobe software Adobe Mixamo* or *FUSE*. This software is very easy to use and practical for people who do not have the experience to create 3D characters. *Mixamo* interface like the one in Figure 4. Using *Mixamo* software we can save the time required to manufacture the NPC characters in the normal size takes a long time.

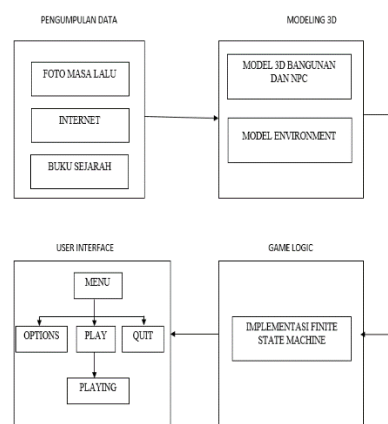


Figure 1. General architecture.

Unity3D is equipped with features *Asset Store* where there are quite a lot of free and paid asset that can be used to build a virtual environment. The virtual environment is a holistic entity of the landscape

that has been designed in accordance with the desired. *Landscape* itself is made using the existing terrain features on Unity3D and refer to the map of Tanjung Pura created by the colonial Dutch East Indies as in Figure 3. The results of the modelling of virtual environments can be seen in Figure 5. For a player author using UFPS v1.5.0 as prototype for First person view.



Figure 2. Front view of the train station tanjung pura.



Figure 3. Map railways tanjung pura – stabat.

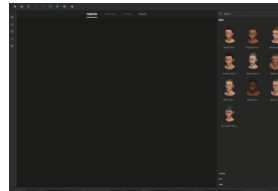


Figure 4. Display adobe fuse.

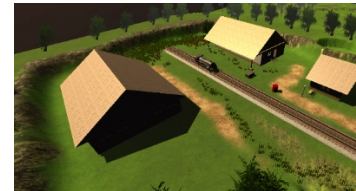


Figure 5. Virtual environment railway station.

2.3. Game Logic

Game logic in this game using the *finite state machine* in the *non-player character* that acts as enemies and allies. Their task is to kill the player or team who shared player. *Game logic* is based on the basic flow of the game. Plot of the game begins with the player selecting a play on the menu interface, when it was in the game the player will see Lt. *Lidan Syam* as NPC that *barking* (cheering), player approached the NPC continues pressing the "F" that is directed to the NPC and appeared dialog box select the next and next to get the mission. After appearing popup player mission press the "ESC" and choose "Quest" keeps track of the quest. It is said to be killed 5 Dutch soldiers. After killing the Dutch army, then the player must meet Lt. *Lidan Sham* back and reported that the mission has been completed can only proceed to the next scene. In the second scene the player will meet Lieutenant Amir Yahya and told me, after the player is required to repel the Dutch troops to pass through the bridge Stabat. Ending the scene in this game and will return to the main menu.

Finite State Machine (FSM) is a model used to describe and regulate the flow of execution is typically used for computer program and sequence logic circuits. It is suitable to be applied against the AI in the game, producing great results without the complex code [7]. [8] For AI programming, authors have the option *deterministic* or *non-deterministic*. Most video games have deterministic. Means you know how the enemy will react differently based on input if they will attack. If you hide and wait, they will attack you. Application of FSM begins with states and transitions required. Imagine FSM in Figure 6 illustrates the brains of ants carrying leaf to home. The starting point is the *state find leaf*, which will be active until the ants find the leaf [7]. When that happens, the condition of state in transition to *go home*, which will be active until the ants reached the nest. When ants to nest, active state into the *find leaf* again, so ants repeat the journey. If the state is active is to find the mouse cursor approaching leaf and ants, the transition into a state *run away*. When the active state, the ants will run from the mouse cursor. When the cursor is no longer a threat, state transition back to *find leaf*. Because there is a transition that connects the *find leaf* and *run away*, the ants will always run away from the mouse cursor when it approaches the ants for ant looking leaf. That will not happen if the current state is *go home* (check Figure 7). In this case the ants will return home without fear, only *find leaf* transition when the ants go to home.

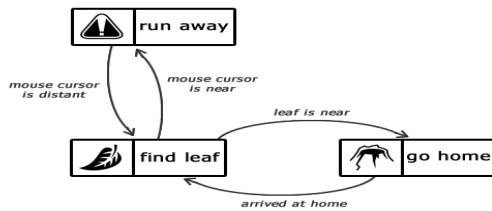


Figure 6. The brain of the ant.

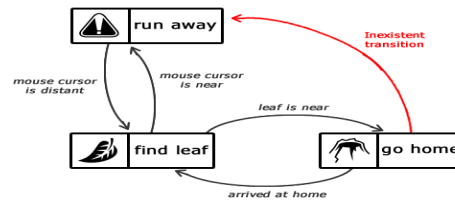


Figure 7. FSM describes the brain of an ant. Note the lack of transition between run away and go home.

In this study, the script will be used as the brains of all activities undertaken by AI is FPSBrain.cs. When in Play, initially enemy will patrol in accordance with a predetermined waypoint. When the enemy saw the approaching players, he would pursue until a specified minimum distance away and engage (shoot) to the player until the player has a 0% Health Point and enemies back to the normal state can be silent or patrol back. Program code as follows:

NPC bit of understanding. NPC known as a character who is not a person or character that is not played, the game is any character that is not played by the player or players. In *video games*, this usually means the character controlled by the computer via *artificial intelligence* (AI). If the *Player Character* (PC) in the story of the protagonist, a NPC can be considered a "supporting player" or "extras" of narrative Role playing. NPC characters to fill the fictional world of the game and can fill any role that is not occupied by the player character. NPC characters may be allies, bystanders or competitors for the player character. NPC also can trader who trades currency for such things as equipment or supplies npc that vary in the level of detail [11].

2.4. User Interface

At this stage it will be designed the *user interface* of the game Battle Stabat. The design of the *user interface* is designing the layout of the display application to be built. *User interface* design must consider several aspects. One of the most important aspects of user interface design is user convenience and high level of user interface that is *user friendly*. The significance of ease of use (*user friendly*) because the purpose of the display is the user interface to allow users to use the application. How applications can be run easily by users who had never run the application as an important factor in the design of the user interface. Flowchart User Interface can be seen in Figure 8.

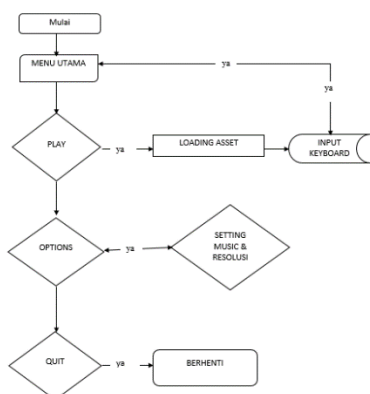


Figure 8. Main menu display design.

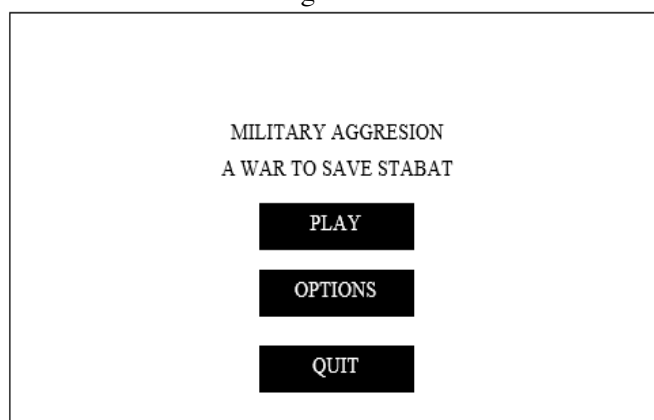


Figure 9. Flowchart system application.

At the start of the game the player will be exposed to the *Main Menu* interface game where there are *Play*, *Options* and *Quit*. If the player chooses *Play*, the system will continue to the next command to *loading asset*, loading the assets used as a hint that the game is running. Once completed, the computer requires input from the player to move forward and backward as well as shooting in the game. On the *Options* menu there is an option to set to music and the screen resolution that can be customized by the player. From the submenu setting music player can return to the main menu by pressing the *back* button. Last is a *Quit* menu to stop playing the game and the system ends.

On the main menu page, there are three main menus. These menus are displayed in a *button*. The main menu is, the *Play* button, the *Options* button, and the *Quit* button. While the *Options* button when pressed it will display the *sub-menu*, which is to set the music volume and size of the resolution used. The main menu display can be seen in Figure 9.

3. Result and Discussion

The game display is a display when running the game start menu on the main page. The first process after running the play menu, the application will display a loading screen so that the system can make the process calling the asset. This game is made with a view *First Person Camera*, where the user becomes the main actor in playing the game. As in Figure 11. Explanation of numbering in Figure 11 are as follows:

3.1. Health Point (Blood) Player.

Health *Point* on the player is a fundamental component of a condition that can make a player off when health point equals 0. Health point the player will not grow when exposed to damage.

3.2. Number of Ammo

The amount of ammo or ammo on the player at the start-up is 19/0 which means has nineteen bullets and bullet zero reserves. By the time play players will get new weapons and ammo after speaking first with NPC Lieutenant Sham Lidan.

Display when the player runs out of HP (*Health Points*) on the image 12 is where the player has the blood of 0% or below 0%, which means the *player is dead*.

3.3. Testing

Testing is needed to assess how well the method used in a system. If the method used does not work as it needs further investigation regarding the factors that influence the occurrence of these constraints. The method was tested at this stage is a method npc for Patrol and Engage (shooting) when looking at the enemy, when there is no one enemy npc will return to silent mode or patrol back. In Figure 13 npc patrol. The pink color indicates npc toward a predetermined waypoint.

The author conducted a test on the system to see how many *frames per second* (FPS) game application fps does look smooth in the eyes of the player in accordance with at least a decent frame for games played. Can be seen in Figure 14.

Statistics can be seen in the game that the *frames per second* of the game is 15 FPS can be said to be low. Because of minimal FPS is 30 FPS to make it look smooth and pleasing to the eye of the player. In Unity there is profiler window that helps to optimize the game. Profiler will tell, the author how much time is spent on the various areas of the game. For example, the profiler will tell the percentage of time spent on rendering, animation or game logic. Authors can play games in the Editor when the profiler is enabled, and the profiler will record performance data. Profiler window then displays the data in the time frame, so we can see the frame or the area spike (spent a lot of time) than others. By clicking anywhere on the timeline, the bottom profiler window displays details for the selected frame visible as in Figure 15. The rendering process increases dramatically when the player starts to reach 15 FPS.

Rendering is the process of data on the camera in Unity to sort layer which can be seen by the camera or the characters in the game in a visual form.

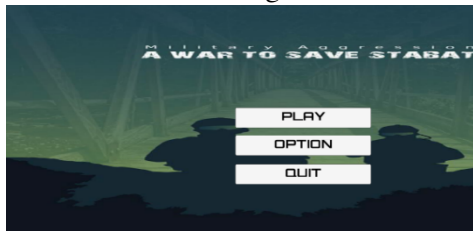


Figure 10. The main menu display



Figure 11. Display in game

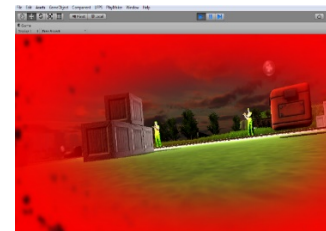


Figure 12. Display player dead



Figure 13. NPC patrolled by waypoint

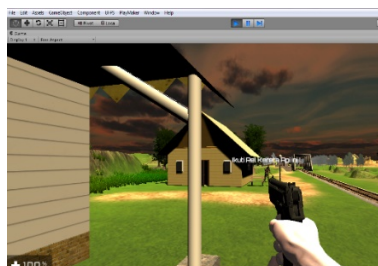


Figure 14. Frame per second this game

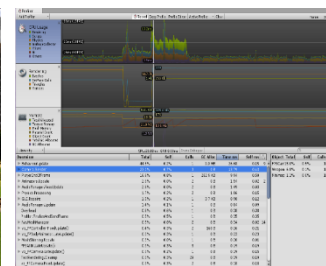


Figure 15. Profiler window unity

4. Conclusions

Sources of information about the war in Stabat is very limited, thus making the 3D models of building and landscape are made only based on existing images, the story of history in research only on an event that occurred war to fight Stabat, so that the continuation of the story is not loaded and only by a very short history book, at the time played this game has a frame rate (fps) of about 15, due to the need of a higher graphics card, and through games we can provide a visual representation of History is happening around us.

Some suggestions that could be considered in future research are simplifying the 3D modeling process, thus rendering more optimally by utilizing real texture, adding more scenes to a story about the history of North Sumatra and extra features NFC as the people who are moving around with utilizing Artificial Intelligence (AI).

Acknowledgement

This research was supported by the Universitas Sumatera Utara. All the faculty, staff members and laboratory technicians of Information Technology Department, whose services turned my research a success.

References

- [1] TWH M 2007 Perjuangan rakyat sumatera utara
- [2] Mizanuddin 2011 Pembuatan game FPS (sudut pandang orang pertama) "operasi LUN INAU" dengan menggunakan Unity
- [3] Raharja W S 2011 Pengembangan aplikasi situs sejarah dalam bentuk peta tiga dimensi interaktif kompleks monumen tugu pahlawan surabaya menggunakan Unity 3D engine.
- [4] Fauzi M, Haki and Rodiah 2013 First person shooter 3D "GamaShoot" dengan blender dan Unity 3D 1 – 6
- [5] Peng X 2014 3D game development with unity A case study: A first-person shooter (FPS) game
- [6] Jailani A Q 2010 Strategi gerak hinder non-player character (NPC) pada game first person shooter (FPS) menggunakan finite state machine (FSM)
- [7] Bevilacqua F 2013 Finite-state machines : theory and implementation
- [8] Warden J 2012 Finite state machines in game development
- [9] Jailani A Q 2010 Strategi gerak hinder non-player character (NPC) pada game first person shooter (FPS) menggunakan finite state machine (FSM)
- [10] Meisadri R and Indriani N 2013 Pembangunan permainan first person shooter 3D alien hunter 1 1 – 6
- [11] Ellison B 2008 Non-player character