

## Enhancing Gaming Engagement through the Integration of Game Design Document and Finite State Machine: A Study on Optimizing Non-playable Character Responsiveness

**Jansen Wiratama<sup>1\*</sup>, Rian<sup>2</sup>, Monika Evelin Johan<sup>3</sup>, Hari Santoso<sup>4</sup>**

[Jansen.wiratama@umn.ac.id](mailto:Jansen.wiratama@umn.ac.id), [Rian0443721@gmail.com](mailto:Rian0443721@gmail.com), [monika.evelin@umn.ac.id](mailto:monika.evelin@umn.ac.id), [hari.santoso8687@gmail.com](mailto:hari.santoso8687@gmail.com)

<sup>1,3</sup> Faculty of Informatics and Engineering, Information System Department, Universitas Multimedia Nusantara, Indonesia

<sup>2,4</sup> Information System Department, STMIK Dharma Putra, Indonesia

---

### **Article Information**

Received : 03 Aug 2023  
Reviewed: 07 Aug 2023  
Accepted : 27 Aug 2023

---

### **Keywords**

Finite State Machine,  
Game Design Document,  
Gaming Experience,  
Non-playable Character,  
Responsiveness.

---

### **Abstract**

Non-playable Character (NPC) is a prominent terminology within the gaming domain. While players do not directly control NPCs, their role significantly influences the gaming experience. Enhancing NPC responsiveness becomes imperative as NPCs' passive nature can lead to monotonous gameplay. To address this concern, a computational model, namely Finite State Machine (FSM), is implemented to elevate NPC responsiveness during interactions with the main characters, whether as adversaries or allies. This research uses the Game Design Document (GDD) methodology to design a survival horror-themed game. The resultant Survival Horror Game undergoes Alpha testing to validate its overall functionality and the successful integration of the FSM computational model. Findings indicate that enemy NPCs can pursue the main character from any position, with an average arrival interval of 68 seconds. Additionally, allied NPCs promptly respond when the main character approaches. Subsequently, the Beta testing results reflect an 80% average score interpretation based on percentage responses, implying the game's favorable suitability and acceptability on the Likert scale.

---

## **A. Introduction**

In game development, there is a term called Non-Playable Character (NPC). NPCs are characters that players cannot control. NPCs contribute significantly to the gaming experience. Objects called NPCs allow players to engage in dialogue, attack if the NPC acts as an enemy, ask for help, and so on [1]. NPC was first coined and introduced 1975 in Dungeons and Dragons [2]. Furthermore, if there are NPCs in a game, it is necessary to apply artificial intelligence so that the NPCs become responsive and adaptive to what the players are doing. The responsiveness of NPC impacts players with an exciting and intense playing experience [3]. A literature study was conducted to obtain relevant references for this study [4]. Previous research has successfully applied the Finite State Machine (FSM) Computational Model for Non-Playable Character Behavior [5]. FSM can also be applied through game design with Unity3D tools [6]. The FSM Computing Model can also be applied to develop NPCs in games with elements of culture and themes in Indonesia [7]. In addition, FSM is also used to design several game genres, such as 2-dimensional horror games [8] and games with the theme of endless runners [9].

Implementation is an expansion of activities that adjust to each other; for that, the implementation of FSM on NPC is needed in game design in this study [10]. FSM implementation is needed so that activities, actions and actions can be planned to increase the responsiveness of the NPC, which is one of the objectives of this study [11]. Increased Responsiveness on NPC correlates with artificial intelligence (AI) [12][13].

Another study found that games are integral to forming a child's personality [14]. In a book entitled "A Theory of Fun for Game Design," games become an interactive experience that gives players patterns about the gameplay that they have learned to become increasingly challenging and can eventually be mastered [15]. Meanwhile, the Survival Horror genre, which was chosen as the theme in this study, is a genre construction that has existed for a long time because it was formed through the experience of fear in real life [16][17][18]. The Survival Horror game in this study was designed with Unity tools and the C# programming language, which is equipped with the main character, and supporting characters in the form of the main character's allies and enemy characters. Initially, the NPC in this game could not act intuitively so that the NPC acting as an enemy could not act responsively to chase the player when the player passed a random time interval between 60-70 seconds resulting in the enemy NPC not moving from an idle position, with x-axis worth 0, the y-axis is worth -0.14, and the z-axis is worth 0. NPCs who act as allies of the main character also experience the same problem, which they cannot act intuitively, such as responding according to the main character's movement; the expected response is to have discussions with the main character and ask for help.

## **B. Research Methodology**

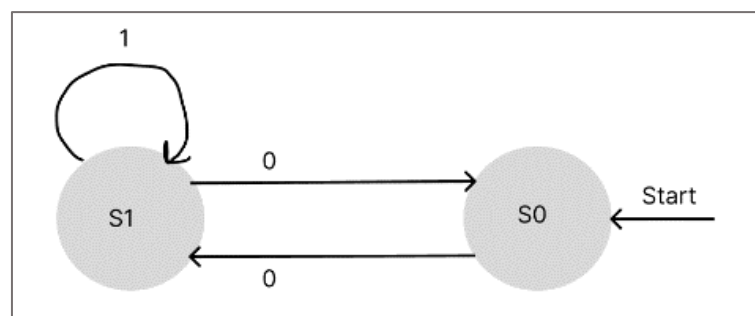
Game Design Document (GDD) is one of the methods included in the Software Design Document category. The process in GDD has many steps but focuses on four different attributes of the program, namely data structure, software architecture, interface representation, and details of the algorithms used [19]. Before the game

enters the design stage, it is necessary to have a descriptive document for designing [20].

Finite State Machine (FSM), also known as Finite State Automata (FSA), is a computational model that can also be used to simulate logical sequences or represent execution sequences and conditions [21]. One example of implementing a Finite State Machine is an automatic vending machine. The Finite State Machine method is also commonly used as a basis for designing communication protocols, designing game software, web applications, and so on [5]. There are two types of Finite State Machines, and it generally has five tuples  $\{Q, \Sigma, q, F, \delta\}$  with the explanation below:

1.  $Q$ : the set of all states  $\{q_0, q_1, q_2, q_3, \dots\}$ .
2.  $\Sigma$ : the set of all input or event symbols  $\{x, y, \dots, 0, 1, \dots, a, b\}$ .
3.  $q$ : initial state, with condition  $(q_0 \in Q)$ .
4.  $F$ : a set of final states where  $(F \subseteq Q)$ .
5.  $\delta$ : transition function, where  $\delta: Q \times \Sigma \rightarrow Q$ .

A simple example of a Finite State Machine representation is as follows:



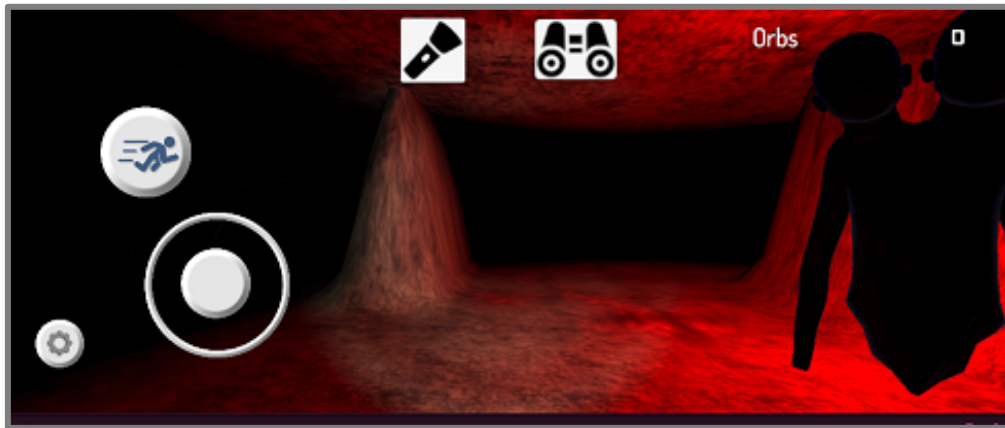
**Figure 1.** Finite State Machine analogy

1.  $M = \{Q, \Sigma, q, F, \delta\}$
2.  $Q = \{S0, S1\}$
3.  $\Sigma = \{0, 1\}$
4.  $q_0 = \{S0\}$
5.  $F = \{S1\}$

The  $M$  symbol represents the set of Finite State Machine tuples. Notation  $Q$  as a set of all states having two elements  $\{S0, S1\}$ ,  $\Sigma$  as a set of events having two elements  $\{0, 1\}$ , initial state/state being  $\{S0\}$ , and set of final states having one element namely  $\{S1\}$ . Alpha Testing was carried out to test the results of the application of FSM in designing this survival horror game, which is a method used to test a game and look for errors in the games that have been made [22]. At the same time, Beta Testing involves users trying already functionally functioning games. Respondents in this beta testing were selected based on their experience and expertise in playing the game [23]. Qualitative research involving interviews with five experienced horror gamers was conducted to gather information for designing the game. Selection of informants considered their expertise and familiarity with interactive games from previous research reviews [24][25][26].

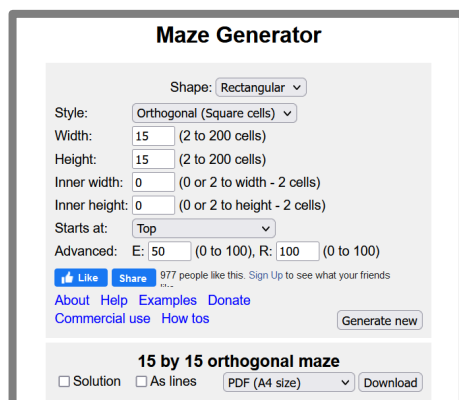
### C. Results and Discussion

Based on the application of the Game Design Document (GDD) method, it generated games that can be played; the appearance of the game is presented with screenshots and explanations below:



**Figure 2.** Gameplay screenshot

The implementation of the labyrinth as a dungeon has various stages. The first stage is creating a maze design using a Maze Generator.



**Figure 3.** Maze Generator settings



**Figure 4.** Maze Generator results

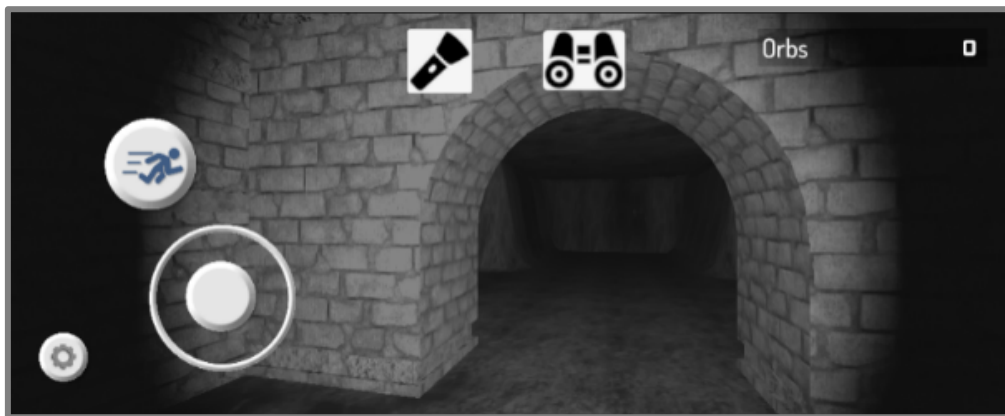
The maze can be made with various customizations, starting from the shape, width, height, inner width, inner height, and so on. In this case, the maze is created to be 15 in width, 15 in height, and orthogonal in style. After the labyrinth design is generated, implementing it into the game project Invading Darkness in Unity is the next step. The maze design will be embedded in an object called terrain using the stamp feature. This stamp will later stamp the labyrinth design on the intended terrain. After completing this process, the final step that must be done is to place the maze into the terrain, resulting in Figure 4.

The button for activating the Night Vision Goggle (NVG) is integrated with scripts, while for creating NVG effects, an effect called post-processing is used. When the NVG button is tapped, the button emits a sound indicating that the tool is active, the entire screen turns gray, and the viewing radius increases by 29.4. Figure 5 shows the game display before NVG was activated. Figure 6 shows the game display before NVG was activated. With the appearance changing to gray and

the visibility getting more precise and farther with the post-processing statistics as follows:

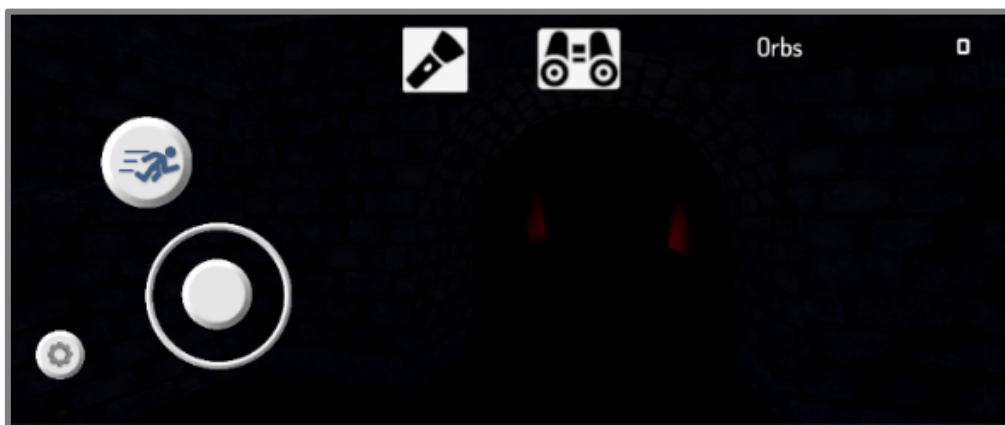


**Figure 5.** Before *NVG* activated



**Figure 6.** After *NVG* activated

The integration of the Flashlight activation button with scripts is similar to the implementation of Night Vision Goggles (NVG). Additionally, the creation of a luminous effect is achieved through the utilization of an object referred to as a spotlight. The visual representation in Figure 3.2 illustrates the appearance of the game prior to the activation of the Flashlight feature.

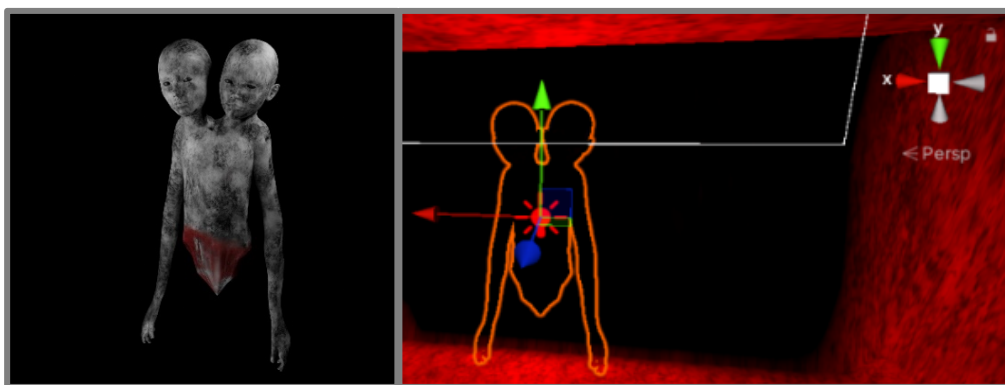


**Figure 7.** Before flashlight activated



**Figure 8.** After Flashlight activated

When the Flashlight button is tapped, the button emits a sound indicating that the device is active, and a light appears with an intensity value of 2.5, a viewing distance of 29.4, and a point angle of 92.7.



**Figure 9.** NPC Enemy three-dimensional model and light object

In this game, the NPC Enemy is depicted as a unique model, resembling two small children conjoined at the body but with separate heads. Their lower body portion features a sharp tip, adorned with blood-like stains. The implementation of this enemy involves integrating a three-dimensional model with a red point light, creating an eerie effect where the children's bodies emit a red glow resembling blood (Figure 9). To govern the behavior of both enemy and allied NPC objects, a Finite State Machine (FSM) computational model is employed during the coding process. Upon successful implementation of the FSM, the enemy NPC exhibits responsive behavior triggered by a random interval of 60-70 seconds. This periodicity adds unpredictability and suspense to the gameplay, enhancing the overall horror experience for players. The testing phase, including alpha testing, represents a critical stage in this research, where the functionality of the game is thoroughly examined. During alpha testing, developers rigorously assess the game's performance, identifying and addressing any bugs, glitches, or balance issues that may impact the players' immersion and enjoyment. By combining the intricate design of the NPC Enemy model with the adaptive nature of the FSM, the

game aims to deliver a gripping and engaging experience. Players can expect a hair-raising encounter with the eerie conjoined children, whose unpredictable actions and haunting appearance will heighten the tension and thrill throughout the gameplay.

**Tabel 1.** Alpha Testing

Features	Description	Results
<b>Navigation Button</b>		
<b>Digital Analog</b>	Moving the main character in various directions can make the main character walk.	Successfully
<b>Sprint</b>	It makes the main character run fast.	Successfully
<b>Settings Button</b>	Access settings for game configuration.	Successfully
<b>Sensitivity Settings</b>	Set the gaming motion sensitivity.	Successfully
<b>Scene</b>		
<b>Main menu</b>	Scene to display the main menu.	Successfully
<b>Gameplay</b>	Scene to start playing.	Successfully
<b>Win</b>	The scene that is displayed if you win the game.	Successfully
<b>Lose</b>	The scene is displayed if the enemy kills the main character.	Successfully
<b>BGM</b>		
<b>MainMenu BGM</b>	The music that plays while in the MainMenu scene	Successfully
<b>Gameplay BGM</b>	The music that plays while in the Gameplay scene	Successfully
<b>Enemy BGM</b>	The music plays while the enemy is chasing the main character.	Successfully
<b>Logics, AI</b>		
<b>Finite State Machine</b>	A computational model is applied to enemy NPCs so they can chase and kill the main character in 60-70 second time intervals.	Successfully
<b>NoteAppearTrigger</b>	The object serves as a trigger to bring up the chronological record of Rosale's kidnapping.	Successfully
<b>NightVision Button</b>	Button to turn Night Vision Toggle on/off.	Successfully
<b>Flashlight Button</b>	Button to turn Flashlight on/off.	Successfully

The scenario used in this FSM is to measure time using a stopwatch, placing the main character's object at a random position to find out whether the implementation of the FSM computational model on enemy NPCs can work or not. Here are the results of the test:

**Tabel 2.** Finite State Machine Testing

<i>i</i>	Gameobject									$\tau$	<i>s</i>
	Main Character	Enemy NPC									
		State									
		Idle				Gameover					
		Position			Position			Position			
x	y	z	X	y	z	x	y	z			
1	0	1,25	0	0	-0,14	0,5	0.35	0.23	0.81	67.8	✓
2	-48.34	0.09	-0.92	0	-0,14	0,5	-0.06	-0.14	0.48	64.3	✓
3	-61.5	0.09	-0.92	0	-0,14	0,5	0	-0.14	0.5	65.1	✓
4	-94.3	0.09	-0.92	0	-0,14	0,5	0	-0.14	0.85	67.2	✓
5	9.2	0.09	8.9	0	-0,14	0,5	0.01	-0.14	-0.09	62.6	✓
6	21.5	-0.31	-15.9	0	-0,14	0,5	-0.21	-0.14	-0.24	63.1	✓
7	-39	-0.31	-15.9	0	-0,14	0,5	3.2	1.74	-9	69.5	✓
8	-61.4	-0.31	-21.2	0	-0,14	0,5	-4.77	1.74	-13.31	66.6	✓
9	-80.76	-0.31	-28.62	0	-0,14	0,5	-11.66	1.74	-19.35	64.2	✓



<i>i</i>	Gameobject									$\tau$	<i>s</i>
	Main Character	Enemy NPC									
		State									
		Idle				Gameover					
	Position			Position			Position				
x	y	z	X	y	z	x	y	z			
10	-102.3	-0.31	-28.62	0	-0,14	0,5	-19.33	1.74	-19.35	68.8	✓
11	-68.66	-0.31	-45.6	0	-0,14	0,5	-7.36	1.74	-33.2	61.3	✓
12	-18.5	-0.31	-45.6	0	-0,14	0,5	10.49	1.74	-33.2	68.2	✓
13	-8.6	-0.31	-9.3	0	-0,14	0,5	14	1.74	-3.7	62.7	✓
14	-44	-0.31	9.3	0	-0,14	0,5	1.4	1.74	11.4	66.2	✓
15	-87.1	-0.31	9.3	0	-0,14	0,5	-13.94	1.74	11.4	61.9	✓
16	-112.7	-0.31	9.3	0	-0,14	0,5	-23.07	1.74	11.4	61.7	✓
17	-112.7	-0.31	-50.9	0	-0,14	0,5	-23.07	1.7	-37.6	63.6	✓
18	-42.8	-0.31	-50.8	0	-0,14	0,5	1.83	1.78	-37.5	68.3	✓
19	-7	-0.31	-50.8	0	-0,14	0,5	14.57	1.78	-37.5	64.2	✓
20	24.9	-0.31	-33.1	0	-0,14	0,5	25.92	1.78	-23.09	64.9	✓
21	-33.7	-0.31	-43.1	0	-0,14	0,5	5.06	1.78	-31.2	68	✓
22	-70.7	-0.31	-43.1	0	-0,14	0,5	-8.11	1.78	-31.2	61.4	✓
23	-70.7	0	-69.6	0	-0,14	0,5	-8.11	1.78	-52.8	67.3	✓
24	-18.2	0	-69.6	0	-0,14	0,5	10.58	1.78	-52.8	63.1	✓
25	3.7	0	-69.6	0	-0,14	0,5	18.37	1.78	-52.8	64.5	✓
26	3.7	0	-111	0	-0,14	0,5	18.37	1.78	-86.5	61.7	✓
27	-50.6	0	-111	0	-0,14	0,5	-1	1.78	-86.5	66.9	✓
28	-50.6	0	-69.8	0	-0,14	0,5	-1	2.09	-53	61.5	✓
29	-27.6	0	-96.3	0	-0,14	0,5	7.2	2.09	-74.6	65.7	✓
30	12.8	0	-96.3	0	-0,14	0,5	21.6	2.09	-74.6	68.4	✓

Notation Description:

1. *i* = iteration; repetition; nth test.
2.  $\tau$  = time; in table 4.3,  $\tau \Rightarrow$  when does the enemy NPC appear in a random interval between 60-70 seconds.
3. *s* = status to indicate whether the test was successful.

In Table 2, testing is carried out by placing the position of the main character object on a random axis position to ensure that the enemy NPC can chase the main character in any position without any problems in the form of a silent NPC not responding when it has passed a random time interval of 60-70 second. The initial position of the enemy NPC is idle, with the x-axis having a value of 0, the y-axis having a value of -0.14, and the z-axis having a value of 0.5. The test results can be seen in the column that explains the position of the enemy NPC in a game over state. In the first to the 30th iteration, it was found that the enemy NPC caught up with the main character and attacked him. In the enemy NPC sub-column, the chasing condition is not included in the table because the chasing condition of the



enemy is moving to and from toward the main character; it causes the value of the axis position to be not constant and is not included in the testing.

Meanwhile, in the game over state, the enemy NPC's axis position is constant because the main character has been killed, and the enemy NPC stops moving, which includes the game over state in the testing table. Chasing state => game over state, meaning game over state will not occur if the chasing state is unsuccessful. The average result of the NPC arrival time interval in this Alpha testing is 68 seconds, indicating the response follows planning.

After doing Alpha testing, the following testing stage is to do Beta testing, where an application is tested directly by the player. This beta testing was conducted to test the feasibility of NPC interactivity when played by players. The number of respondents in Beta testing is as many as seven people. Each respondent has a background as someone who has played Horror games. Respondents will be asked to play this game and then fill out an online questionnaire to obtain information about the tester's experience regarding the game. The questionnaire contains 15 statements that seven testers must fill in. The following is a list of the 15 statements:

**Tabel 3.** Statement List's

No	Statements
1	The game is scary.
2	The game is easy to play.
3	The buttons on this game work as they should.
4	UI (User Interface) in the game looks attractive.
5	The game's UX (User Experience) is simple and straightforward for the player.
6	Orbs object works fine (disappears when hit by the player).
7	The number of total orbs increases as orbs are collected.
8	The button to turn on Night Vision works fine.
9	The button to turn on the Flashlight works fine.
10	NPCs acting as main character allies work well. (when approaching the NPC, text appears on the screen explaining the state of the NPC)
11	NPCs acting as enemies work fine. (Can penetrate building objects; within a specific time, can chase the main character; the main character dies if touched by the NPC)
12	The background music used in the game gives a scary/intimidating impression.
13	The labyrinth gives a scary and intimidating impression because of its slightly "crowded" or narrow shape.
14	The dark lighting level in Resurrection of Darkness gives a scary impression.
15	You will want to play this game again in the future.

In the questionnaire used in the Beta testing facility, the scoring parameter uses a Likert scale which has five scales in the following format: 1=Strongly disagree, 2=Disagree, 3=Neutral, 4=Agree, 5=Strongly agree. The scale determines the level of agreement with a series of statements regarding the experience of playing this game. The following is a summary of the data obtained from each tester who filled out the questionnaire:

**Table 4.** Questionnaire Recapitulation

Age	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
23	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4
18	4	3	2	3	3	4	3	5	5	5	5	5	5	5	4
22	5	3	5	4	5	5	5	5	5	5	5	5	5	5	4
18	5	5	4	4	5	5	5	5	5	5	4	4	5	5	5
23	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
21	4	4	5	4	3	4	4	5	5	5	4	5	5	5	4

21 4 4 5 4 4 4 5 4 5 5 4 4 5 4 4

### Abbreviation:

1. Age = Respondent age
2. S-n = n statement; example: S15 = 15<sup>th</sup> statement.

The first stage first calculates the total score of each respondent with the T x Pn formula, where T means the total number of respondents selected and Pn is the Likert score number choice. After calculating the score, the next step is to interpret the calculated score to find out the score with the highest and lowest scores. The score with the highest score is represented by the symbol (X), and the score with the lowest score is represented by the symbol (Y); the following is the calculation:  $X = 5 \times 7 = 35$  and  $Y = 1 \times 7 = 7$

After these calculations, it can be seen that the highest score for X is 35, and Y is 7. Next is the pre-completion stage to find out the interval and interpretation of the percent with the following formula:

$$\text{Index (\%)} = \frac{\sum \text{skor}}{X \times 100}$$

**Tabel 5.** Total score results

Statements	Respondent answer					Total Score	Results in percentages
	1 (STS)	2 (TS)	3 (N)	4 (S)	5 (SS)		
1	-	-	-	4x4 = 16	3x5 = 15	31	$\frac{31}{35 \times 100} = 88.5\%$
2	-	-	2 x 3 = 6	2x4 = 8	3x5 = 15	29	$\frac{29}{35 \times 100} = 82.8\%$
3	-	1 x 2 = 2	-	2x4 = 8	4x5 = 20	30	$\frac{29}{35 \times 100} = 82.8\%$
4	-	-	1x3 = 3	5x4 = 20	1x5 = 5	28	$\frac{28}{35 \times 100} = 80\%$
5	-	-	2x3 = 6	2x4 = 8	3x5 = 15	29	$\frac{29}{35 \times 100} = 82.8\%$
6	-	-	-	4x4 = 16	3x5 = 15	31	$\frac{31}{35 \times 100} = 88.5\%$
7	-	-	1x3 = 3	2x4 = 8	4x5 = 20	31	$\frac{31}{35 \times 100} = 88.5\%$
8	-	-	-	2x4 = 8	5x5 = 25	33	$\frac{33}{35 \times 100} = 94.2\%$
9	-	-	-	1x4 = 4	6x5 = 30	34	$\frac{34}{35 \times 100} = 97.1\%$
10	-	-	-	1x4 = 4	6x5 = 30	34	$\frac{34}{35 \times 100} = 97.1\%$
11	-	-	-	4x4 = 16	3x5 = 15	31	$\frac{31}{35 \times 100} = 82.8\%$
12	-	-	-	3x4 = 12	4x5 = 20	32	$\frac{32}{35 \times 100} = 91.4\%$
13	-	-	-	1x4 = 4	6x5 = 30	34	$\frac{34}{35 \times 100} = 97.1\%$
14	-	-	-	2x4 = 8	5x5 = 25	33	$\frac{33}{35 \times 100} = 94.2\%$
15	-	-	-	5x4 = 20	2x5 = 10	30	$\frac{30}{35 \times 100} = 85.7\%$

In the Likert scale, the criteria for interpreting scores in the interval are:

1. 0% – 19,99% = Very poor
2. 20% – 39,99% = Poor
3. 40% – 59,99% = Acceptable
4. 60% – 79,99% = Good
5. 80% – 100% = Very good

Based on the percentage results obtained, the average percentage of all statements is 80%, which means very good and worth playing.

#### **D. Conclusion**

The game was successfully designed with increased responsiveness in interacting with the main character, either as an enemy or an ally, proven by Alpha testing that all aspects function properly and the Finite State Machine computational model can be applied with the test results that the enemy NPC can chase the main character in any position with an average arrival time interval of enemy NPCs is 68 seconds and allied NPCs can respond if the main character approaches. In beta testing, the average interpretation of the score in the form of a percentage of tester answers to all beta testing questionnaire statements is 80%, which implies that the game is worth playing.

#### **E. Acknowledgement**

We extend our heartfelt gratitude to Universitas Multimedia Nusantara for their invaluable support, which played a pivotal role in the successful completion of this research endeavor. Their substantial contribution was instrumental in achieving our objectives, and we are deeply grateful for their unwavering assistance.

#### **F. Reference**

- [1] H. Warpefelt, "The Non-Player Character-Exploring the believability of NPC presentation and behavior," 2016.
- [2] Inworld, "What is an NPC? Tracing the history (and future) of NPC meaning in games, memes, and Tik Tok," 2022. <https://www.inworld.ai/blog/what-is-an-npc-tracing-the-history-and-future-of-npc-meaning-in-games-memes-and-tik-tok> (accessed May 20, 2023).
- [3] R. D. Fajri, M. A. Syahputra, T. R. M. Zaki, A. Saifudin, and I. Kusyadi, "Perancangan Kecerdasan Buatan pada NPC Menggunakan Unity 2D dan Perilakunya terhadap Player," vol. 6, no. 3, pp. 2622–4615, 2021, doi: 10.32493/informatika.v6i3.11843.
- [4] Salmaa, "Studi Literatur: Pengertian, Ciri-Ciri, dan Teknik Pengumpulan Datanya," <https://penerbitdeepublish.com/studi-literatur/>, 2023. <https://penerbitdeepublish.com/studi-literatur/> (accessed Mar. 23, 2023).
- [5] S. R. Hermawan, "Penerapan Metode Finite State Machine Pada Game 'The Mahasiswa' Guna Membangun Perilaku Non Playable Character," 2018. Accessed: May 19, 2023. [Online]. Available: [https://dspace.uui.ac.id/bitstream/handle/123456789/12528/14523032\\_LaporanSkripsi.pdf?sequence=1](https://dspace.uui.ac.id/bitstream/handle/123456789/12528/14523032_LaporanSkripsi.pdf?sequence=1)
- [6] M. F. Rahadian, A. Suyatno, and S. Maharani, "Penerapan Metode Finite State Machine Pada Game 'The Relationship,'" 2016.
- [7] B. Basrie, J. D. Hasiholan, and A. R. Suteja, "Implementasi Algoritma Finite State Machine Sebagai Perubahan Perilaku Otonom Dan Adaptif Pada Non-Player Character Dalam Game Petualangan Ksatria Pancasila," *Sebatik*, vol. 25, no. 1, Jun. 2021, doi: 10.46984/sebatik.v25i1.1252.

- [8] N. Mardiyani, "Skripsi Penerapan Metode Finite State Machine Dalam Perancangan Perilaku Musuh Pada Game 2d Horror Girl's Nightmare Program Studi Teknik Informatika S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang 2018," 2018.
- [9] M. A. Mahendranta, E. Muhammad, A. Jonemaro, and M. A. Akbar, "Penerapan Decision Making NPC dengan Metode Hierarchical Finite State Machine Pada 2D Endless Runner Game," 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [10] Z. Ahmad, "Penerapan Dua Bahasa (Arab dan Inggris) dalam Komunikasi Antar Santri di Pondok Pesantren Al-Kautsar Pekanbaru," 2017. Accessed: Jun. 04, 2023. [Online]. Available: <https://repository.uir.ac.id/778/>
- [11] D. P. Kevin, "Penerapan Pelatihan Penggunaan Alat-Alat Keselamatan Oleh Crew Mv. Baruna Maju," 2021.
- [12] N. Kritiana. Z, "Pembangunan Sistem Pakar Pengobatan Penyakit Menggunakan Obat Tradisional Suku Dayak Berbasis Web," 2015.
- [13] V. Amrizal and Q. Aini, *Kecerdasan Buatan*. Jakarta, 2013. Accessed: May 20, 2023. [Online]. Available: <https://repository.uinjkt.ac.id/dspace/bitstream/123456789/44538/2/naskah%20kecerdasan%20buatan.pdf>
- [14] A. Budianto, "Analisis Dan Perancangan Game Edukasi 'Need For Safety' Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun, Yogyakarta," 2014.
- [15] J. Gregory, *Game Engine Architecture*. 2018. [Online]. Available: <http://taylorandfrancis.com>
- [16] S. Loutzenhiser, "THE DECAY OF MUSUHS: HORROR MOVIES THROUGHOUT HISTORY," 2016. Accessed: Mar. 23, 2023. [Online]. Available: <https://core.ac.uk/download/pdf/80854016.pdf>
- [17] E. Fazrin, "Wanita Dan Tato: Studi Eksploratif Pencarian Sensasi Pada Wanita Pengguna Tato Di Kota Bandung," *Penerapan Sistem Bisnis Keuangan dalam Mendukung Society 5.0*, 2019, Accessed: Jun. 04, 2023. [Online]. Available: <http://repository.unibi.ac.id/138/1/WANITA%20DAN%20TATO%20STUDI%20EKSPLORATIF%20PENCARIAN%20SENSASI.pdf>
- [18] D. Masson, "How does the dark sound?," 2020, Accessed: Jun. 04, 2023. [Online]. Available: <https://shs.hal.science/halshs-03139600>
- [19] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. 2014. Accessed: Mar. 22, 2023. [Online]. Available: [https://repository.dinus.ac.id/docs/ajar/Software\\_Engineering\\_-\\_Pressman.pdf](https://repository.dinus.ac.id/docs/ajar/Software_Engineering_-_Pressman.pdf)
- [20] R. Putratama, "Perancangan Game Design Document Untuk Game Cockatoo Defender," 2020. Accessed: Mar. 22, 2023. [Online]. Available: [https://ifik.telkomuniversitas.ac.id/assets/upload/thesis/rafiipp/PERANCANGAN\\_GAME\\_DESIGN\\_DOCUMENT\\_UNTUK\\_GAME\\_COCKATOO\\_DEFENDER3.pdf](https://ifik.telkomuniversitas.ac.id/assets/upload/thesis/rafiipp/PERANCANGAN_GAME_DESIGN_DOCUMENT_UNTUK_GAME_COCKATOO_DEFENDER3.pdf)
- [21] A. Taru, "Penerapan Finite State Machine Pada Perancangan Game | Berita | Gamelab Indonesia," 2020. <https://www.gamelab.id/news/206-penerapan-finite-state-machine-pada-perancangan-game> (accessed May 19, 2023).
- [22] Leonardo, J. Pragantha, and D. A. Haris, "Jurnal Ilmu Komputer dan Sistem Informasi PEMBUATAN GAME HACK AND SLASH 2D 'SERENADE TOWER,'" no. Vol. 8 No. 2 (2020): Jurnal Ilmu Komputer dan Sistem Informasi, 2020, Accessed: Jun.

- 04, 2023. [Online]. Available:  
<https://journal.untar.ac.id/index.php/jiksi/article/view/11533>
- [23] Startup Studio Indonesia, "Memahami Apa Bedanya Beta Testing & Alpha Testing | SSI," <https://startupstudio.id/beta-testing/>, 2021. <https://startupstudio.id/beta-testing/> (accessed Jun. 26, 2023).
- [24] A. Heryana, "Informan Dan Pemilihan Informan Dalam Penelitian Kualitatif," 2018.
- [25] S. Hansun, "Scratch Pemrograman Visual untuk Semuanya", *Ultima InfoSys : Jurnal Ilmu Sistem Informasi*, vol. 5, no. 1, pp. 41-48, Jun. 2014.
- [26] P. Altanny and M. Johan, "Web-based Decision Support System for Characters Selection in Game Genshin Impact with SAW Method", *Ultima InfoSys : Jurnal Ilmu Sistem Informasi*, vol. 14, no. 1, pp. 40-51, Aug. 2023.