JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 10.30811/jtrik.v8i1.7422
Vol. 8 No. 1 2025 pp. 63-69

Implementation of Fuzzy Logic in NPCs for Online Multiplayer Games Using Photon

Muhammad Haris[1], Mursyidah[2]*, Atthariq[3]

[1,2,3] Department of Information and Computer Technology, Politeknik Negeri Lhokseumawe, Indonesia

Coresponding Author: mursyidah@pnl.ac.id

Article info: Received 03/02/2025, Revised 10/02/2025, Accepted 04/07/2025

Abstract

This thesis discusses the implementation of fuzzy logic techniques in managing the behavior of non-playable characters (NPC) in a game with online multiplayer features that utilize the Photon platform. Games with online multiplayer modes have become very popular, but managing NPC behavior in a multiplayer context is often a complex challenge, requiring high adaptability to changes in the environment and player behavior. This research describes the process of implementing fuzzy logic on NPCs to improve interactivity and player experience. in multiplayer online games. There are three main components in this study: the NPC behavior model, the selection of relevant environmental variables, and the design of fuzzy rules for decision making. In addition, this thesis also describes the integration of Photon technology as a multiplayer platform, which allows players to interact in real-time in the game.The experimental and evaluation results show that the use of fuzzy logic in managing NPC behavior in multiplayer online games can increase the level of realism in player interactions with NPC characters. This makes a positive contribution to the player's gaming experience and results in a more challenging and exciting gameplay in an online multiplayer context. This research provides valuable insights into the use of fuzzy logic in online multiplayer games as well as the integration of Photon as an effective multiplayer platform.

Keywords: first, second, three, four, five

1. Introduction

Side-scrolling games began in the arcade era in the late 1970s and early 1980s. At that time, the technology used was still limited, but side-scrolling games had become a popular genre. Some classic side-scrolling games from that period included "Space Invaders" (1978), "Defender" (1980), and "Donkey Kong". The development of side-scrolling games presents unique challenges that developers must face. Overcoming these challenges requires deep understanding of game design, development skills, and creative thinking abilities from the development team. With thorough planning, careful testing, and player feedback, developers can address these challenges and create engaging and satisfying side-scrolling experiences.

In previous research, artificial intelligence was implemented using NPCs merely for interaction purposes, specifically in game processes that only involved defeating NPCs. After successfully defeating NPCs, collectible items would be obtained. The interaction performed by NPCs was limited to attacking the player. Therefore, the authors applied the fuzzy logic algorithm to an action and educational genre game called "7 Wonders of the World" in 2D format. This game can interact with NPCs in a non-monotonous manner.

The "7 Wonders of the World" game uses health parameters. Fuzzy logic operates on health parameters, where when the parameter falls below 200, the NPC will interact by increasing attack strength and speed. Fuzzy logic also works on player attacks affecting the NPC's health bar parameter

A. Side-scrolling

Side-scrolling refers to a type of video game or software application displayed from a side-view perspective, where the player controls a character or object that moves from the left side of the screen to the right, or vice versa. Characters typically move across platforms, explore environments, and battle enemies or avoid obstacles. Side-scrolling games initially appeared on arcade and classic video game consoles, such as Mario Bros, Sonic the Hedgehog, and Contra. However, this genre has now been adapted to various platforms,

including mobile games and computer software.[1]

B. Multiplayer

Multiplayer is a feature of a game that allows players to interact with one another. It makes games more interactive and enjoyable. Multiplayer games are divided into offline and online multiplayer. Offline multiplayer allows players to interact without the need for an internet connection, such as "Two Guys & Zombies" (using Bluetooth) and "Ludo King" (played together on the same device). In contrast, online multiplayer requires an internet connection and offers the advantage of being playable from any distance.[2]

C. NPC

A Non-Player Character (NPC) is a character in a video game controlled by the computer, not by players. NPCs can serve as supporting or antagonistic characters, provide information or quests, or act as merchants within the game. In some games, NPCs can also be allies assisting the player in completing tasks or missions. NPCs can be programmed to move automatically or interact with players in various ways, enriching the gameplay experience by adding narrative depth, context, and interaction.[3]

D. 2D Game

A 2D game is a type of video game that uses two-dimensional representation (x and y axes) for graphics and gameplay. In 2D games, the environment and characters are displayed in a flat perspective, similar to an image on paper. Character or object movement is usually restricted to two dimensions (horizontal and vertical). 2D games have a long history and remain popular for their simple design, strong gameplay, and nostalgic appeal.[4]

E. Inspector

The Inspector is a crucial panel in the Unity development interface. It is located on the right side and is used to edit the properties and components of selected objects in the Hierarchy or Scene. Using the Inspector panel, developers can easily adjust and configure object properties and their components without editing code directly.

F. Animator Controller

The Animator Controller in Unity is part of the animation system used to control character and object animations. It allows for smooth transitions between animations based on input, conditions, and events, giving greater depth to gameplay.[5]

G. Animation Character

Animation Character in Unity refers to the process and techniques for animating characters (2D or 3D) within a game or application environment. It involves making characters move, speak, and interact with their environment realistically or as required by the design.[6]

H. Asset

Assets refer to the various resources or elements used in game or application development, such as environments, characters, objects, sounds, music, animations, scripts, textures, and more. Assets are the building blocks of Unity projects and are essential for creating rich user experiences.[7]

2. Methods

A. Use Case Diagram Design

A Use Case Diagram is a visual representation depicting the interactions between users (actors) and a system or application. It shows various scenarios of how users will interact with the system and the expected outcomes, providing an overview of the main functionalities to be used.[8]

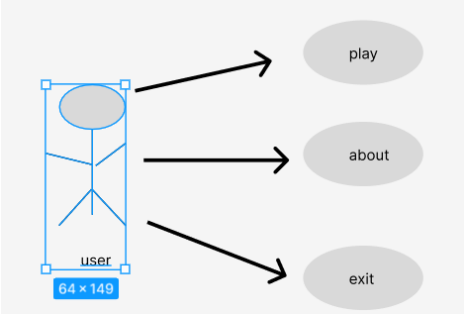

Figure 1. Use Case Diagram

Figure 1 shows the menu interface at the initial stage of the game application.

B. Fuzzy Logic Design

The research method can be illustrated in a flowchart.

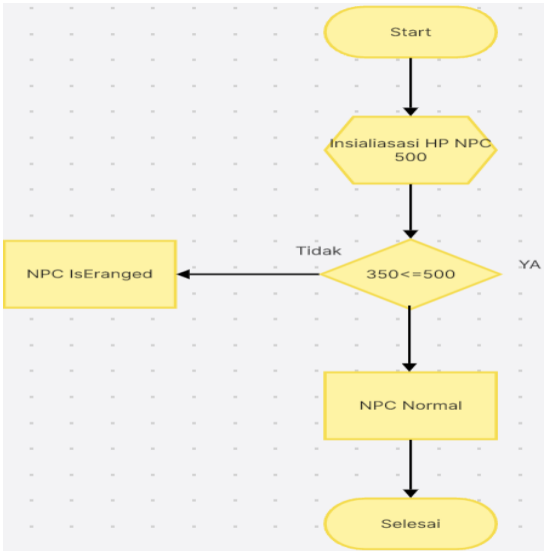

Figure 2. Fuzzy Logic Flowchart

The flowchart begins with a terminator symbol, followed by a preparation step for initializing constants—specifically, setting the NPC's HP to 500. A decision block checks if HP is between 350 and 500. If true, the process sets the NPC state to normal; if not, it sets the NPC state to isEnranged. The flowchart ends with a terminator labeled "finished".[9]

3. Result and Discussions

A. Software Design

1) Main Menu Display

The game's initial screen displays the Main Menu, including several buttons such as "Play" to start the game, "About" for game information, and "Exit" to quit.

Figure 3. Main Menu Display

2) Gameplay Display

The gameplay screen shows the selected scene and the play area. It also includes the Heads-Up Display (HUD), which provides information such as NPC health, character assets, and background.

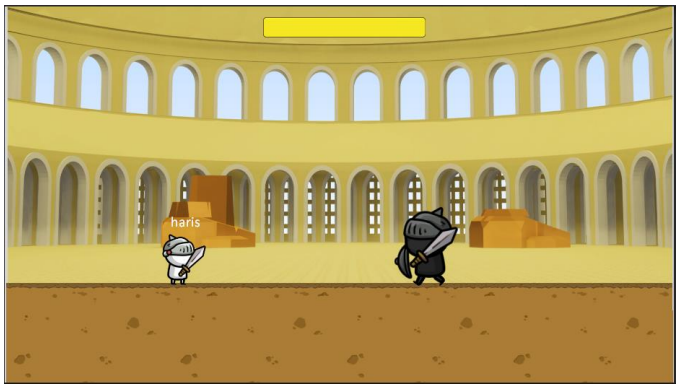

Figure 4. Gameplay Display

B. Fuzzy Algorithm Implementation

Fuzzy logic is a method where condition checking, in this case the NPC's HealthBar, indicates the NPC's state changing from normal to isEnraged condition. The implementation steps based on the Flowchart of the fuzzy logic algorithm mentioned previously in chapter 3 are as follows.

1) Fuzzification

In Fuzzification, there are several variables. Table I shows these variables.

TABLE I. MEMBERSHIP FUNCTION VARIABLES

Variabel	**Set**
NPC Health	High, Medium, Low
Damage	Weak, Strong
Action (Output)	Normal, Enraged

In Table I, each Variable column contains sets from their respective membership functions.

2) Health Membership Function

In the membership function, there is a health variable that indicates ranges. Ranges can be seen in Table II below.

TABLE II. HEALTH MEMBERSHIP FUNCTION

Health Variable	**Range**
Low	0-50
Medium	100-200
High	350-500

Table II explains that the health membership function contains variables ranging from low to high.

3) Damage Membership Function

In the membership function, there is a damage variable that indicates ranges. Ranges can be seen in Table III below.

TABLE III. DAMAGE MEMBERSHIP FUNCTION

Damage Variable	**Range**
Weak	20-25
Medium	30-35
Strong	40-45

C. Test Results

The next step is to test the membership functions above to determine the fuzzy rules contained within them. Table IV shows the results of fuzzy logic experiments.

TABLE IV. FUZZY LOGIC EXPERIMENT RESULTS

Test ID	Health Range	Damage Range	Action Range
P1	Many (300%)	Weak (20%)	Normal
P2	Many (450%)	Weak (20%)	Normal
P3	Many (325%)	Weak (20%)	Normal
P4	Few (100%)	Strong (40%)	IsEnranged
P5	Medium (150%)	Strong (40%)	IsEnranged
P6	Medium (170%)	Strong (40%)	IsEnranged

Table IV shows that the Fuzzy Logic algorithm generally succeeds in determining NPC behavior. Based on test results, the fuzzy logic algorithm successfully determines NPC behavior.

D. Fuzzy Rules

- Rule 1: IF NPC Health is high, THEN NPC Damage = 20
- Rule 2: IF NPC Health is healthy, THEN NPC Damage = 20
- Rule 3: IF NPC Health is healthy, THEN NPC Damage = 20
- Rule 4: IF NPC Health is medium, THEN NPC Damage = 40
- Rule 5: IF NPC Health distance is medium, THEN NPC Damage = 40
- Rule 6: IF NPC Health is medium, THEN NPC Damage = 40

E. Defuzzification

1) Normal Health Set Membership Function

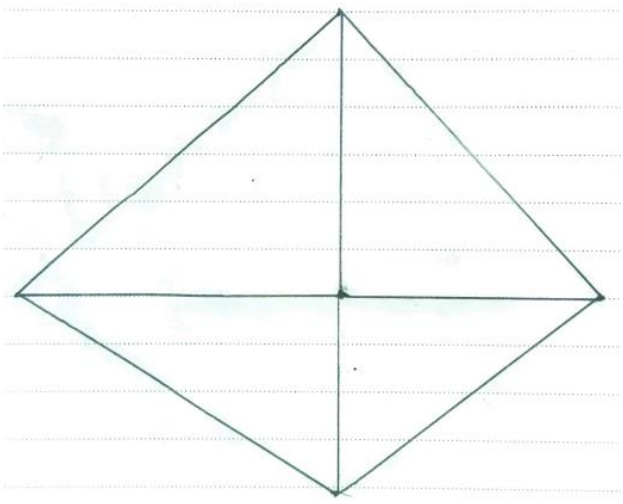

Figure 5. Normal Health Set Display

The above display shows the object form resulting from the normal health set with coordinates 250, 250, 350, 500 in trapezoidal shape.

2) Enraged Health Set Membership Function

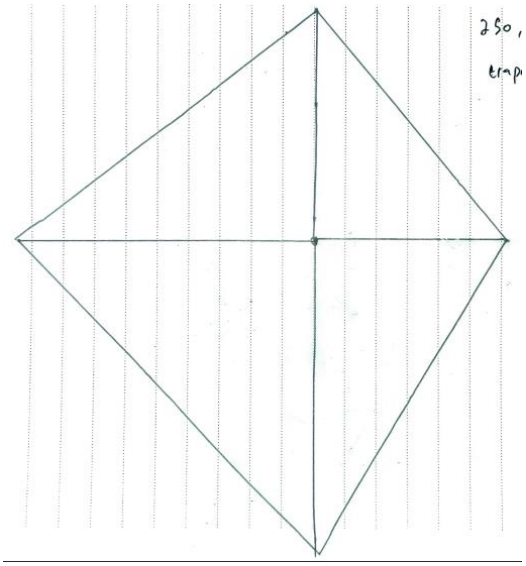

Figure 6. Enraged Health Set Display

The above display shows the object form resulting from the enraged health set with coordinates 150, 200, 210, 250 in trapezoidal shape.

3) Normal Action Set Membership Function

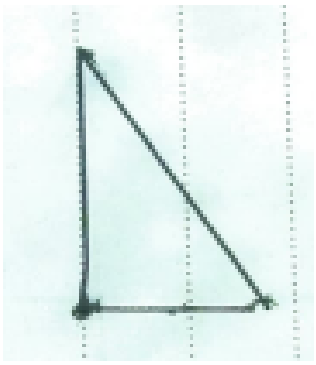

Figure 7. Normal Action Set Display

The above display shows the object form resulting from the normal action set with coordinates 0, 1, 2 in triangular shape.

4) Enraged Action Set Membership Function

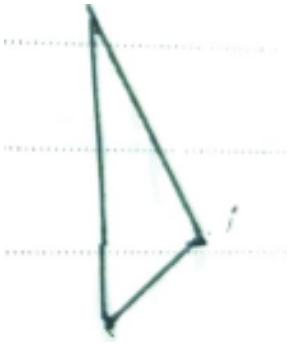

Figure 8. Enraged Action Set Display

The above display shows the object form resulting from the enraged action set with coordinates 1, 2, 3 in triangular shape.

Rule 1: If NPC_Health is "Normal" and NPC_Attack is "Normal", then NPC_Action is "Normal".

Rule 2: If NPC_Health is "Enraged" and NPC_Attack is "Enraged", then NPC_Action is "Enraged" [10].

4. Conclusion

Based on the research conducted by the authors, it can be concluded that the implementation of fuzzy logic on NPCs in games was successfully accomplished with the following conclusions:

1. This research successfully developed a design for a 2D online multiplayer Side Scrolling game.
2. The implementation of the Fuzzy Logic algorithm in this game has increased the level of adaptability and realism in the gaming experience.
3. The results of this research show increased interactivity levels in online multiplayer games resulting from the use of the Fuzzy Logic algorithm.

REFERENCES

[1] S. R. Haryono, U. Ciputra, and U. C. Town, "Rancang Bangun Side Scrolling Action Role - Playing Game Bertema Budaya Surabaya Menggunakan Unity Berbasis Android," *Informatika*, pp. 129–136, 2017.

[2] A. Maulana and R. Firmansyah, "Pembangunan Game Petualangan Platformer Ivira," vol. 2, no. 2, 2021.

[3] A. S. Milak, E. W. Hidayat, and A. P. Aldya, "Penerapan Artificial Intelligence pada Non Player Character Menggunakan Algoritma Collision Avoidance System dan Random Number Generator pada Game 2D 'Balap Egrang,'" *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 5, p. 985, 2020, doi: 10.25126/jtiik.2020711816.

[4] N. Ramsari and G. Ramadhan, "Pembuatan Game Side Scrolling 2D the Naila'S Survival Berbasis Android," *J. FIKI*, vol. VIII, no. 2, pp. 2087–2372, 2018, [Online]. Availablehttp://jurnal.unnur.ac.id/index.php/jurnalfiki

[5] C. S. Winter, "Assignment 2 : Animations in Unity," no. 1, pp. 1–4, 2018.

[6] R. Y. Ramadhani and A. Qoiriah, "Implementasi Algoritma Fuzzy Sugeno sebagai Pendukung Keputusan dalam Penentuan Skor Akhir pada Game Edukasi Simple Nomic," *J. Informatics Comput. Sci.*, vol. 1, no. 01, pp. 48–55, 2019, doi: 10.26740/jinacs.v1n01.p48-55.

[7] Eni, "済無No Title No Title No Title," *Angew. Chemie Int. Ed. 6(11), 951–952.*, no. Mi, pp. 5–24, 1967.

[8] R. A. Ardenia and A. E. Budianto, "Penerapan Metode Fuzzy Untuk Game Keselamatan Pengendara Sepeda Motor Berbasis Android," 2016, [Online]. Available: https://www.neliti.com/publications/184524/penerapan-metode-fuzzy-untuk-game-keselamatan-pengendara-sepeda-motor-berbasis-a

[9] A. Dijkstra and F. L. Sugeno, "Penerapan algoritma dijkstra dan fuzzy logic sugeno pada game zombie shooter," *Core.Ac.Uk*, no. x, 1978, [Online]. Available: https://core.ac.uk/download/pdf/80765954.pdf

[10] S. V Lidianto, G. S. Budhi, and R. Intan, "Perancangan dan Pembuatan Action Game dengan Artificial Intelligence dan Machine Learning," *J. Infra*, 2016, [Online]. Available: http://publication.petra.ac.id/index.php/teknik-informatika/article/view/4500%0Ahttps://publication.petra.ac.id/index.php/teknikinformatika/article/download/4500/4125