**Recursive Journal of Informatics**
Vol. 3, No. 1, March 2025
p-ISSN 2963-5551 https://journal.unnes.ac.id/journals/index.php/rji e-ISSN 2986-6588

# Application of Fuzzy Logic in Visual Novel Evaluation System Using Unity 3D

**Epafraditus Memoriano[1], Riza Arifudin[2]**
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

**Purpose:** Visual novels, narrative-driven games focused on character interaction, commonly employ point-based evaluation systems that struggle to represent the inherent complexity and uncertainty of player choices. This research introduces a novel approach: integrating fuzzy logic into visual novel evaluation systems using Unity 3D. Fuzzy logic addresses the limitations of point-based systems by accounting for the "fuzzy" nature of player choice and its varied impact on story progression and character relationships.
**Methods/Study design/approach:** A visual novel game was developed in Unity 3D, incorporating a fuzzy logic evaluation system for scoring player choices and assessing route progress. Fuzzy sets and membership functions were defined for key aspects like emotional response, character alignment, and plot development. These aspects were dynamically evaluated based on player dialogue selection, and individual scores were aggregated to generate a final route evaluation.
**Result/Findings:** Testing demonstrated seamless integration of the fuzzy logic system within the game engine. Evaluation of conversation choices and route progression yielded accurate and nuanced scores, reflecting the varying weight of each decision based on narrative context and character interaction. Fuzzy logic facilitated the interpretation of "fuzzy" player choices, translating them into meaningful information for story progression and character relationships.
**Novelty/Originality/Value:** This research presents a novel and promising approach to visual novel evaluation by leveraging the strengths of fuzzy logic. It overcomes the limitations of traditional point-based systems, capturing the complexity and dynamism of player choices within the narrative. The dynamic and responsive evaluation results enhance player engagement and provide a more immersive gaming experience.

**Keywords**: Fuzzy logic, visual novel, game evaluation, player choice, narrative complexity, Unity 3D
**Received** February 02, 2024 / **Revised** July 04, 2024 / **Accepted** March 27, 2025



## INTRODUCTION

In the realm of visual novels, a unique gaming experience unfolds, where players actively shape the narrative through decision-making, influencing the storyline and desired outcomes. However, the creation of visual novels presents challenges in crafting optimal difficulty levels and gameplay experiences to cater to diverse player preferences. To tackle these challenges, the application of fuzzy logic emerges as an intriguing and innovative approach to optimize the assessment system and game dynamics. Fuzzy logic allows the incorporation of variables and rules that accommodate fuzzy or interval values, enabling the assessment system to operate beyond the binary "true" or "false" paradigm [1]. Leveraging fuzzy logic empowers the assessment system to evaluate various factors with a degree of ambiguity, resulting in more dynamic, adaptive, and engaging visual novel games [2]. In the context of developing visual novels using Unity, a prominent game development platform [3], integrating fuzzy logic becomes not only feasible but also more straightforward.

Unity provides a range of tools and capabilities to construct interactive gaming environments with flexibility in configuring gameplay mechanisms [4], along with robust support for fuzzy logic component integration. This opens doors for game developers to create more complex and sophisticated assessment systems capable of better responding to player interactions. While the use of fuzzy logic in the context of visual novel games serves as an early exploration model, it offers insights into the effective application of fuzzy logic in various real-life situations and scenarios [5].Nevertheless, research on the implementation of fuzzy logic in the creation of visual novel games using Unity faces several challenges. The development

[1]*Corresponding author.
Email addresses: epamemo@students.unnes.ac.id (Memoriano)
DOI: 10.15294/rji.v3i1.1158

and implementation of assessment systems using fuzzy logic may require in-depth and complex analysis in determining the appropriate variables and rules to reflect the desired level of ambiguity [6]. Additionally, careful evaluation and testing are necessary to ensure the accuracy and performance of the implemented assessment system. Hence, this research involves a meticulous development and iteration process to achieve the expected results.

To contextualize this research within the existing body of knowledge, a literature review reveals insights from previous studies. Several researchers have explored the application of fuzzy logic in game development, with Soedargo and Junaedi [1] shedding light on its potential in introducing ambiguity to decision-making processes. Krisdiawan [2] further emphasize the role of fuzzy logic in creating dynamic and adaptive gaming experiences. Despite these advancements, a limited number of studies have delved into the integration of fuzzy logic specifically in the context of visual novels, highlighting a gap in the existing research. This research aims to fill this void by providing a comprehensive exploration of fuzzy logic's application in enhancing the player experience in visual novel games. While some researchers have focused on the broader application of fuzzy logic in gaming, there is a conspicuous absence of studies addressing its integration into visual novel games. Existing research has primarily concentrated on generic game development, leaving a gap in understanding how fuzzy logic can be harnessed to tailor the interactive narrative experience unique to visual novels. Therefore, this research intends to bridge this gap by focusing on the integration of fuzzy logic in Unity-based visual novel game development.

In summary, with limited studies addressing the application of fuzzy logic in Unity-based visual novel games, this research pioneers a comprehensive exploration in this domain. By leveraging the capabilities of Unity and fuzzy logic, the study aims to enhance the adaptability and dynamism of visual novel games, contributing novel insights to the intersection of game development, fuzzy logic, and interactive narrative experiences. The objectives of this research include the development of a sophisticated fuzzy logic-based assessment system and the evaluation of its impact on player engagement and satisfaction.

## METHODS

This research leverages the established Game Development Life Cycle (GDLC) to develop a visual novel game. To enhance narrative dynamism based on player choices, it incorporates a novel fuzzy logic evaluation system within the Unity 3D engine. This system dynamically assesses player decisions based on various factors, impacting the emotional response of characters, story alignment, and plot progression, ultimately enriching the gaming experience with greater nuance and immersion.

### Game Development Life Cycle

Game Development Life Cycle (GDLC) method, which is included in game development, consisting of the stages of planning (initialization), design (pre-production), development (production), testing (testing), optimization (beta) and launch (release) [7]. These stages need to be done to produce a game that can be played by players smoothly and can continue to be developed. The GDLC flow chart can be seen in Figure 1.

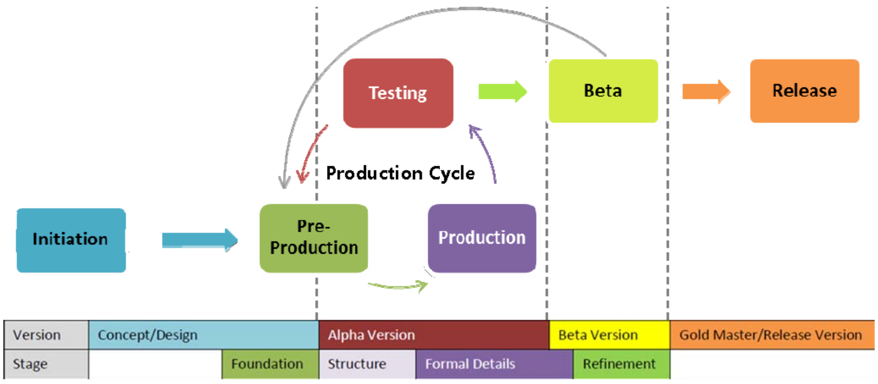


Figure 1. Game Development Live Cycle Flow

In this research, the GDLC method was used to develop a visual novel game using Unity 3D. Based on the feedback from the testing phase, the game was optimized to improve the overall experience. The game was then released to the public.

The implementation of GDLC in this study involved the following stages:

1. Planning: The initial stage in which the game's goals, scope, and requirements are determined. Planning involves identifying the concept, determining the features, and selecting the development platform (Unity 3D).
2. Design: The design stage involves the detailed design of the game, including the design of the characters, backgrounds, interface, and plot of the visual novel. In addition, the design also includes the creation of fuzzy sets and fuzzy rules that will be used in the evaluation system.
3. Development: At this stage, the developers begin to implement the design that has been created into Unity 3D. This includes the creation of scenes, characters, dialogue, and the integration of Mamdani fuzzy logic.
4. Testing: After development is complete, the game will be tested to ensure that all features are working properly. Testing will include general functionality testing as well as testing the validity of the implementation of fuzzy logic.
5. Optimization: After testing, the game will be optimized to improve its performance and quality. This includes visual refinement, improved interaction, and bug reduction.
6. Launch: After the game is considered ready, the game will be launched to the specified platform (e.g. PC or mobile device). Launch involves the distribution and promotion of the game to potential players. Maintenance: After launch, maintenance is performed to fix bugs that may appear after launch and respond to feedback from players.

The game flowchart shows the flow of the game from start to finish. The game starts with the main menu, where the player can choose to play the game, change the settings, or quit. If the player chooses to play the game, they will be taken to the introduction level. In the introduction level, the player can interact with characters and make decisions that will affect their score. The score is used to determine the ending of the game. If the player's score is high, they will get the good ending. If the player's score is low, they will get the bad ending. After the introduction level, the player will progress through the game, making decisions and interacting with characters. The game ends when the player reaches the final level. The flow of the game is visually represented in Figure 2.

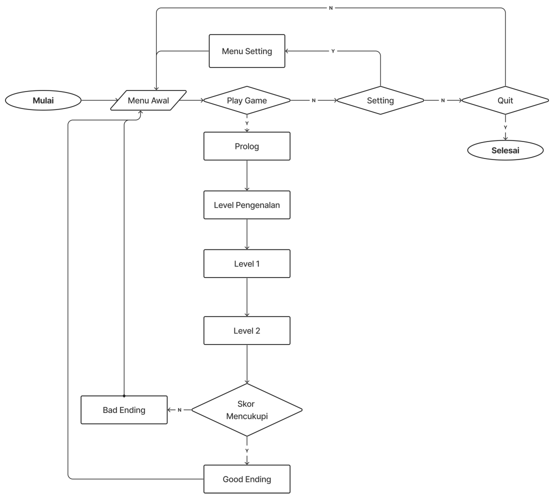


Figure 2. Flowchart for The Visual Novel Game

**Mamdani Fuzzy Logic**

Mamdani fuzzy logic is one approach to fuzzy logic that is used to deal with uncertainty and ambiguity in complex systems [8]. In the context of its application to the evaluation system of visual novel games, Mamdani fuzzy logic can provide a framework that allows the modeling of evaluation based on criteria that have a non-definite degree of truth [9]. Mamdani fuzzy logic involves the concepts of fuzzy sets, fuzzy variables, fuzzy rules, and fuzzy inference [2]. Here is a more detailed explanation of these concepts:

1. Fuzzy Set: A fuzzy set is a set whose elements have a membership degree (truth value) that can be between 0 and 1 [1]. For example, in the context of a game evaluation system, we can have a fuzzy set "Bad Quality" with a membership value that varies depending on the defined criteria.
2. Fuzzy Variable: A fuzzy variable is a variable whose value can be in the form of a fuzzy set [10]. In a game evaluation system, variables such as "Graphics", "Story", and "Gameplay" can be represented as fuzzy variables with the relevant fuzzy sets.
3. Fuzzy Rule: A fuzzy rule is a fuzzy logic statement that defines the relationship between fuzzy variables [11]. These rules are often formulated in the form "If [variable A] is [set A] and [variable B] is [set B], then [output variable] is [output set]".
4. Fuzzy Inference: Fuzzy inference is the process of generating a conclusion or output based on fuzzy rules and given input data [12]. In a game evaluation system, fuzzy inference can be used to combine fuzzy rules with input membership values to produce output membership values [13].

Here are membership functions for Route variables in Figure 3 and membership function for Conversation variables in Figure 4.

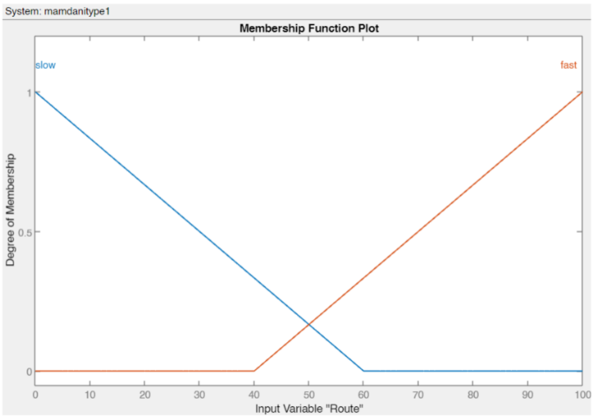


Figure 3. Membership function for Route variables

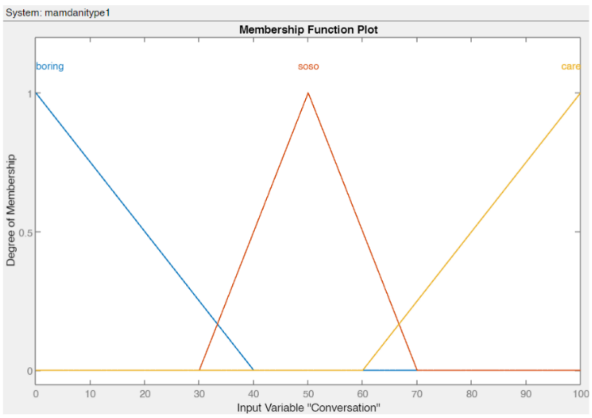


Figure 4. Membership function for Conversation variables

The process of applying Mamdani fuzzy logic involves the following steps:

1. Definition of Fuzzy Sets: Define the relevant fuzzy sets for the input and output variables [12]. For example, in a game evaluation system, we can have the fuzzy sets "Low", "Medium", and "High" for the variable "Conversation".
2. Input Measurement: Take the input value from the system and determine the degree of membership of the input in the fuzzy sets that have been defined [14].
3. Fuzzy Rule Formulation: Create fuzzy rules based on fuzzy logic that define the relationship between the input and output variables [15]. These rules reflect the knowledge and policy in the system.
4. Fuzzy Inference: Use the fuzzy rules and the input membership degrees to generate the output membership degrees [16].
5. Defuzzification: Convert the output membership value into a crisp value that can be used as the final result. Defuzzification methods such as the centroid method or the largest method are used to do this conversion [5].

Example of the processing of input variables using fuzzy Mamdani will look like in Figure 5.

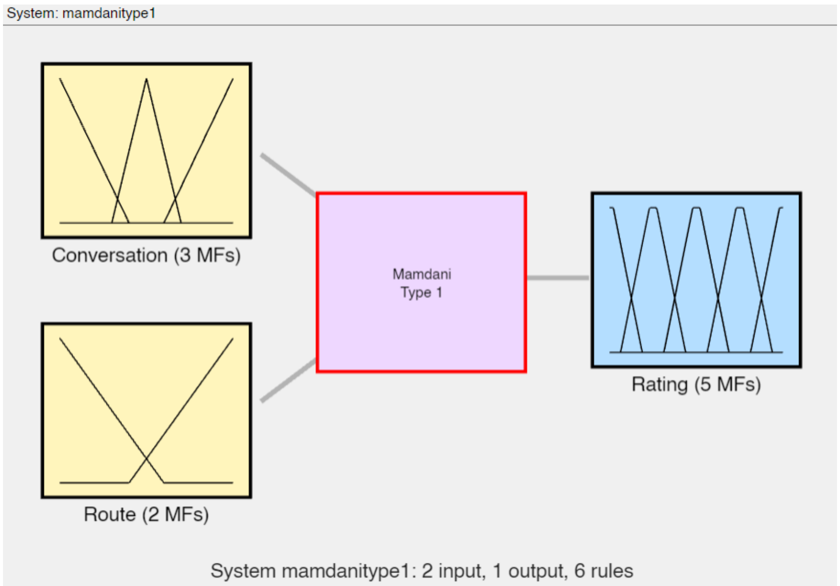


Figure 5. Mamdani Fuzzy System

**RESULT AND DISCUSSION**

The application of fuzzy logic in visual novel game evaluation systems is not just about programming the logic itself, but also about integrating it seamlessly into the overall gaming experience. This requires a deep understanding of narrative, game mechanics, and player psychology, as well as technical skills in software development. The overall process creates an evaluation system that is not only sophisticated and adaptive, but also improves the overall quality of visual novel games.

In this game, there are two inputs: conversation, which determines the quality of the conversation chosen by the player, and route, which evaluates the route taken by the player when taking the customer to their destination. The output parameter generated from this fuzzy logic is Rating with 5 membership functions. The output form that will be displayed to the player will be in the form of stars with a range of 1 to 5 stars

based on the conversion value that has been processed through fuzzy logic. The graph of the input membership function, conversation, is divided into three inputs: boring, so-so, and care.
Based on Table 1, it shows the description of the membership functions of conversation with 3 membership functions, namely boring with domain (0-40) with parameters (0,40), so-so with domain (30-70) with parameters (30,50,70), and care with domain (60-100) with parameters (60,100).

Table 1. Membership Function Values for Conversation

| Membership Functions | Domain | Parameters |
|---|---|---|
| *Boring* | 0-40 | [0,40] |
| *So-so* | 30-70 | [30,50,70] |
| *Care* | 60-100 | [60,100] |

For the input membership function "route," it is categorized into two inputs: "slow" and "fast." As indicated in Table 2, the membership functions for "route" consist of two divisions, specifically "slow" with a domain of (0-60) and parameters (0,60), and "fast" with a domain of (40-100) and parameters (40,100).

Table 2. Membership Function Values for Route

| Membership Functions | Domain | Parameter |
|---|---|---|
| *Slow* | 0-60 | [0,60] |
| *Fast* | 40-100 | [40,100] |
| Membership Functions | Domain | Parameter |

Regarding the output membership function "rating," it is segmented into five inputs: 1 star, 2 stars, 3 stars, 4 stars, and 5 stars. As outlined in Table 3, the membership functions for "rating" comprise five divisions, namely 1 star with a domain of (0 - 18.75) and parameters (0, 2, 18.75), 2 stars with a domain of (6.25 - 43.75) and parameters (6.25, 22.92, 27.08, 43.75), 3 stars with a domain of (31.25 - 68.75) and parameters (31.25, 47.92, 52.08, 68.75), 4 stars with a domain of (56.25 - 93.75) and parameters (56.25, 72.92, 77.08, 93.75), and 5 stars with a domain of (81.25 - 100) and parameters (81.25, 97.92, 100).

Table 3. Membership Function Values for Rating

| Membership Functions | Domain | Parameter |
|---|---|---|
| 1 stars | 0-18.75 | [0,2,18.75] |
| 2 stars | 6.25-43.75 | [6.25,22.92,27.08,43.75] |
| 3 stars | 31.25-68.75 | [31.25,47.92,52.08,68.75] |
| 4 stars | 56.25-93.75 | [56.25,72.92,77.08,93.75] |
| 5 stars | 81.25-100 | [81.25,97.92,100] |

Within the fuzzy logic function that has been established, it is necessary to determine the rules that will be run to process the input. So by setting the right rules, the appropriate results will be obtained. The rules in fuzzy logic used in this study are as follows:

1. If Conversation is boring and Route is slow then Rating is 1 star.
2. If Conversation is so-so and Route is slow then Rating is 2 stars.
3. If Conversation is care and Route is slow then Rating is 3 stars.
4. If Conversation is boring and Route is fast then Rating is 3 stars.
5. If Conversation is so-so and Route is fast then Rating is 4 stars.
6. If Conversation is care and Route is fast then Rating is 5 stars.

Here is the function for calculating the player's score without using the fuzzy logic function. In this source code, the initial stage will activate the panel to display the rating results by changing the parameter

RatingPanel.setActive() to true. After activating the display, you need to get the player's score by storing ResultManager.playerScore.

```
public void RatingStoryPanel()
    {
        ratingPanel.SetActive(true);
        int score = ResultManager.playerScore;
        if (score < 5)
        {
            starAnimator.Play("bintang 1");
        }
        else if (score >= 5 && score < 10)
        {
            starAnimator.Play("bintang 2");
            commentText.text = ratingComment[0];
        }
        else if (score >= 10 && score < 14)
        {
            starAnimator.Play("bintang 3");
            commentText.text = ratingComment[1];
        }
        else if (score >= 14 && score < 17)
        {
            starAnimator.Play("bintang 4");
            commentText.text = ratingComment[2];
        }
        else if (score >= 18)
        {
            starAnimator.Play("bintang 5");
            commentText.text = ratingComment[3];
        }
    }
```

Figure 6. Source Code for *RatingStoryPanel() Function*

The source code in Figure 6 uses a range-based check of the player's score. If the player gets a range of values that meets the criteria in the RatingStoryPanel() function, then the stars or results displayed are in accordance with the function's condition calculation.

```
   public void FuzzyEngines()
    {
        fuzzyEngineDeclare = new FuzzyEngine();
        //=================== INPUT =====================//
        conversation = new LinguisticVariable("Conversation");
        conversation.MembershipFunctionCollection.Add(new MembershipFunction("boring",
0, 0, 0, 40));
```

```
        conversation.MembershipFunctionCollection.Add(new MembershipFunction("so-so",
30, 50, 50, 70));
        conversation.MembershipFunctionCollection.Add(new MembershipFunction("care", 60,
100, 100, 100));

        route = new LinguisticVariable("Route");
        route.MembershipFunctionCollection.Add(new MembershipFunction("slow", 0, 0, 0,
60));
        route.MembershipFunctionCollection.Add(new MembershipFunction("fast", 40, 100,
100, 100));

        //==================== OUTPUT =====================//
        rating = new LinguisticVariable("Rating");
        rating.MembershipFunctionCollection.Add(new MembershipFunction("1", -18.75, -
2.083, 2.083, 18.75));
        rating.MembershipFunctionCollection.Add(new MembershipFunction("2", 6.25, 22.92,
27.08, 43.75));
        rating.MembershipFunctionCollection.Add(new MembershipFunction("3", 31.25,
47.92, 52.08, 68.75));
        rating.MembershipFunctionCollection.Add(new MembershipFunction("4", 56.25,
72.92, 77.08, 93.75));
        rating.MembershipFunctionCollection.Add(new MembershipFunction("5", 81.25,
97.92, 102.1, 118.7));

        //=================== Rules =====================//
        fuzzyEngineDeclare.FuzzyRuleCollection.Add(new FuzzyRule("IF (Conversation IS
boring) AND (Route IS slow) THEN Rating IS 1"));
        fuzzyEngineDeclare.FuzzyRuleCollection.Add(new FuzzyRule("IF (Conversation IS
boring) AND (Route IS fast) THEN Rating IS 3"));
        fuzzyEngineDeclare.FuzzyRuleCollection.Add(new FuzzyRule("IF (Conversation IS
so-so) AND (Route IS slow) THEN Rating IS 2"));
        fuzzyEngineDeclare.FuzzyRuleCollection.Add(new FuzzyRule("IF (Conversation IS
so-so) AND (Route IS fast) THEN Rating IS 4"));
        fuzzyEngineDeclare.FuzzyRuleCollection.Add(new FuzzyRule("IF (Conversation IS
care) AND (Route IS slow) THEN Rating IS 3"));
        fuzzyEngineDeclare.FuzzyRuleCollection.Add(new FuzzyRule("IF (Conversation IS
care) AND (Route IS fast) THEN Rating IS 5"));
        fuzzyEngineDeclare.LinguisticVariableCollection.Add(conversation);
        fuzzyEngineDeclare.LinguisticVariableCollection.Add(route);
        fuzzyEngineDeclare.LinguisticVariableCollection.Add(rating);
        fuzzyEngineDeclare.Consequent = "Rating";
    }
```

Figure 7. Source Code for *Fuzzy*Engines() Function

The source code in Figure 7 is the use of the FuzzyEngine library, which is re-declared based on the fuzzy functions needed in the game. In the FuzzyEngines() function, it is necessary to declare the input membership functions, output membership functions, and also the rules that will be used in processing the input.

The application of fuzzy logic in the visual novel game rating system starts with the creation of the visual novel game using Unity 3D. In the process of creating a visual novel game, the menus that will appear and interact with the player are compiled. In the game, when the player is faced with a situation where they have to choose a conversation to talk about with another character, the dialog choice display becomes a key aspect of gameplay. The display for these dialog choices, as shown in Figure 8, is designed to give the player a clear overview of the conversation options available to them.

Figure 8. Game Display When Choosing A Conversation

In this display, each dialog choice is typically presented in a format that is easy to read and intuitive, allowing the player to understand their choices quickly. The display often includes clearly written dialog text, as well as possible visual indicators that indicate the mood or tone of each choice. This allows the player to make informed decisions about how they want to interact with other characters, which is essential in visual novel games where conversation choices can have a significant impact on the story's course and the game's ultimate outcome. If the player has completed the level, a rating menu will be displayed based on the player's interactions in that level. The following is the level rating display with a rating that can be seen in Figure 9.

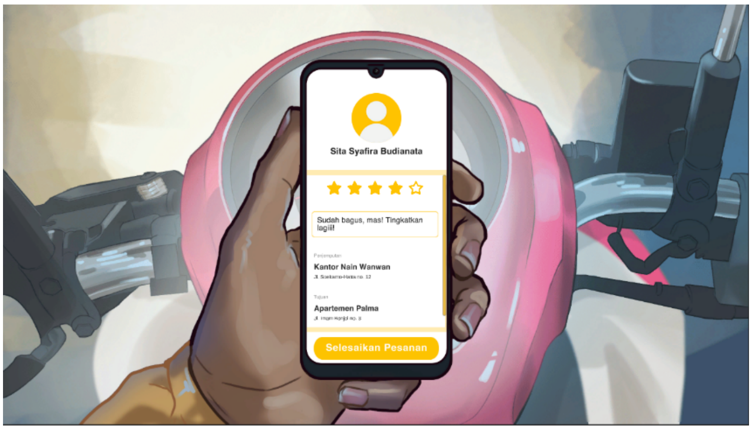

Figure 9. Level Rating Display

Testing visual novel games with a fuzzy logic-based assessment system has provided significant insights, aiming to evaluate the effectiveness of fuzzy logic in assessing player actions. The results indicate successful integration, yielding accurate and reflective scores. Further assessments, including conversation choices and route evaluations, demonstrated the system's ability to process fuzzy values into detailed scores, capturing nuances compared to traditional binary systems.

Throughout testing, the dynamic and responsive fuzzy logic assessment system created a immersive gaming experience, where player decisions had varied impacts on scores, depending on interactions with the story and characters. Identified constraints, such as the lack of a storage system, offer opportunities for future research. Implementing a sophisticated storage system could enhance the gaming experience, ensuring story continuity and player engagement.

Future research could focus on refining the fuzzy logic assessment system, exploring advanced algorithms or integration with technologies like artificial intelligence. The results highlight the potential of fuzzy logic in creating a richer gaming experience, confirming its applicability in visual novel games and suggesting possibilities for improvement and further development.

## CONCLUSION

Based on the research findings and discussions regarding the Implementation of Fuzzy Logic in the Assessment System of Visual Novel Games Using Unity 3D, the following conclusions can be drawn. Firstly, fuzzy logic can be successfully applied to the assessment system of visual novel games by adopting the Game Development Life Cycle method and placing the evaluation system at the end of the level to determine the storyline that players will follow, ensuring the smooth flow of the game. Secondly, the implementation of fuzzy logic in the assessment system of visual novel games results in different responses to the choices made by players, leading to varying impacts on the final outcomes. This dynamic application makes the gameplay more intricate, enhancing the complexity and providing different nuances to each gaming experience.

## REFERENCES


[1] D. S. O. Soedargo and H. Junaedi, "Dynamic Difficulty Adjustment Berbasis Logika Fuzzy Untuk Procedural Content Generation Pada Permainan Roguelike," *Teknika*, vol. 11, no. 2, pp. 98–105, 2022, doi: 10.34148/teknika.v11i2.468.

[2] R. A. Krisdiawan, M. F. Rohmana, and A. Permana, "Pembuatan Game Runaway From Culik Dengan Algoritma Fuzzy Mamdani," *Buffer Informatika*, vol. 6, no. 1, pp. 33–40, 2020.

[3] D. Murtikah and Djuniadi, "Pengembangan Aplikasi Percakapan Bahasa Inggris Dasar Berbentuk Visual Novel Menggunakan Unity 3D," *JURNAL DIGIT Vol.*, vol. 6, no. 1, pp. 1–10, 2016, [Online]. Available: http://jurnaldigit.org/index.php/DIGIT/article/view/34/20

[4] S. Nasution, A. H. Nasution, and A. L. Hakim, "Pembuatan Plugin Tile-Based Game Pada Unity 3D," *It Journal Research and Development*, vol. 4, no. 1, pp. 46–60, 2019, doi: 10.25299/itjrd.2019.vol4(1).3517.

[5] Y. Rizki, H. Mukhtar, and M. D. Andikarama, "Penerapan Metode Logika Fuzzy dalam mengolah pilihan pada Game Visual Novel Informasi Penerimaan Mahasiswa Baru Fakultas Ilmu Komputer Universitas Muhammadiyah Riau sebagai Sarana Evaluasi Mandiri," *Jurnal Fasilkom*, vol. 10, no. 1, pp. 42–53, 2020, doi: 10.37859/jf.v10i1.1914.

[6] H. Anra and H. Sastypratiwi, "Perancangan Game Visual Novel sebagai Alat Ukur Tingkat Social Anxiety Disorder Remaja," vol. 9, no. 1, pp. 67–75, 2023.

[7] J. Lasmana Putra and C. Kesuma, "Penerapan Game Development Life Cycle Untuk Video Game Dengan Model Role Playing Game," *Computer Science (CO-SCIENCE)*, vol. 1, no. 2020, pp. 1–1, 2021, [Online]. Available: http://jurnal.bsi.ac.id/index.php/co-science

[8] A. Manan, V. Wiley, and T. Lucas, "Programmer Selection Using Modified Fuzzy Mamdani Method," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, p. 108, Aug. 2019, doi: 10.24843/lkjiti.2019.v10.i02.p05.

[9] M. Marbun, H. T. Sihotang, and N. V. Marbun, "Perancangan sistem perencanaan jumlah produksi roti menggunakan metode fuzzy mamdani," *Jurnal Mantik Penusa*, vol. 20, no. 1, pp. 48–54, 2016, [Online]. Available: http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/211/123

[10] E. Wardiana, U. H. Medan, J. H. M. Joni, and C. No, "Fuzzy Sugeno Dalam Menentukan Kelayakan Konsumen Dalam Membeli Properti," pp. 256–262, 2020.

[11] I. I. Shagianto, G. W. Wiriasto, D. F. Budiman, and N. M. Seniari, "Aplikasi Game berbasis Andorid 2D dengan Logika Fuzzy pada NPC (Non-Player Character)," *Journal of Electrical Engineering and Information technology*, vol. 1, no. 1, pp. 41–56, 2023.

[12] D. Rumalowak, Y. A. Lesnussa, and F. Y. Rumlawang, "Penerapan Logika Fuzzy Metode Mamdani Untuk Menentukan Jumlah Produksi Ayudes (Studi Kasus: CV. Abadi Tiga Mandiri Ambon)," *Pattimura International Journal of Mathematics (PIJMath)*, vol. 2, no. 1, pp. 25–32, Apr. 2023, doi: 10.30598/pijmathvol2iss1pp25-32.

[13] E. Pangkatodi, "Implementasi Rule Base System dan Fuzzy Logic Artifical Intelligence pada Game Kartu Capsa," *Infra*, vol. 4, no. 1, pp. 1–7, 2016.

[14] A. N. Putri and A. V. Crissano, "Fuzzy Controller Application For Player Movement On 3D Engine Games Based Smart Agent," *Transformatika*, vol. 15, no. 1, pp. 53–59, 2017.

[15] M. A. Yaqin, S. Naja, S. K. Al-Azhar, and K. Mahbullah, "Penerapan Metode Fuzzy Sugeno pada Analisis RFM untuk Menentukan Indeks Produk pada Permainan Hayday," *Prosiding SENIATI*, vol. 5, pp. 50–56, 2019.

[16] F. Hafizh, Y. Maulita, and H. Khair, "Penerapan Logika Fuzzy Logic Pada Enemy Ai Game Horor 3D the Gate of Nightmare Menggunakan Aplikasi Unity3D," *Jikstra*, vol. 02, no. 02, pp. 86–94, 2020.