International Journal of Engineering & Technology, 7 (4.44) (2018) 222-227

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Sugeno Fuzzy for Non-Playable Character Behaviors in a 2D Platformer Game

Ridwan Rismanto[1]*, Rudy Ariyanto[2], Awan Setiawan[3], Melcinsiasih Elinggar Zari[4]

[1,2,3,4]Department Of Information Technology, State Polytechnic of Malang, Indonesia
**Corresponding author E-mail: rismanto@polinema.ac.id*

Abstract

One aspect for a platformer game to be challenging to play is the NPC (Non Playable Character). An NPC is often placed as an obstacle for player to finish at each level. However, an effort must be done to create a behavior of an NPC to be challenging enough and not monotonous. Creating random behavior is one way. But the problem with this approach is a less interactive NPC behavior because of its inability to respond from the game state accordingly. In this paper, we propose an implementation of Sugeno Fuzzy to create behaviors for the NPCs. This methodology takes player's state and NPC's state to determine what action will be done. The inputs are player to NPC distance, NPC's health and player's weapon ammunition. The outputs are retreat, defense and attack. We applied this methodology in an educational game about Rubela virus vaccination "Healthy Hero" for its boss enemies. Evaluation shows that by applying Sugeno Fuzzy, the NPC behavior can responds to player's state and its own state. The real-time execution resulted in 100% correct behavior according to the predefined rules, therefore increasing the behavior interactivity for the NPC.

Keywords: *Sugeno Fuzzy, Platformer game, Unity Game Engine, Artificial Intelligence*

1. Introduction

In a platformer game, Non Playable Character (NPC) is an obstacle placed in each level along with other obstacles like wall, cliff, poisonous items, and many other objects. Obstacles intentionally placed so the game will feel more challenging and not boring. The NPC or so called the enemy, often has its own animations and behaviors.

There are many methods to control enemy's behaviors, for example by running it in sequences. This approach will results in a predictable enemy action. Other method is to randomly executing action at a given time. This approach is better from the sequential actions because its ability to eliminate the predictability. Both of those approaches missed interactivity and does not count any input or reasoning for the action that will be taken. A well known method is to create some kind of artificial intellegence (AI) that takes player's state and enemy's state to decide what action will be taken by the enemy.

The AI in games have two core objectives, which are play well or play believably (human-like) [1]. So here, the AI is used to control the NPC as if it has its own mind or like being controlled by a human. The most crucial aspects of developing games lies in modeling and predicting individual behavior, using model-based approaches as a standard [14].

In a modelling of intelligent game agents, fuzzy logic brings many benefits. One of the benefit is the simplicity of its formulation. In the other side, the simplicity of fuzzy logic can be a powerful AI technique to achieve complex behavior with simple rules [2].

Fuzzy Sugeno is one kind of algorithm that can be used to apply artificial intellegence for an NPC [13]. In this research, we apply Fuzzy Sugeno to determine NPC action by taking account of player's state and enemy's state. The aim for this methodology is to create interactive action based on those inputs.

Previous works in implementation of Sugeno Fuzzy controls the NPC weapon change in First Person Shooter (FPS) game based on two inputs: enemy distance and amount of player in a team [3]. Another research implements Sugeno Fuzzy to control the enemy behavior in Action-RPG games, which consists of 3 kind of character. Each character used 2 inputs, for example, the archer, the inputs are health and distance [4].

This research aims to increase the interactive behavior of the enemies based on the state of enemy and player. For a testbed, we use an educational platformer game about Rubela virus vaccination, aimed for children, called the "Healthy Hero". The NPC to be applied with this methodology is the boss enemy. We use 3 inputs, which are player to NPC distance, NPC's health and player's weapon ammunition. The outputs are retreat, defense and attack.

This game was developed for Android platform using Unity engine and has 2 dimensional style. The fuzzy are implemented in the game engine using C# programming language.

2. Literature Review

2.1. Sugeno Fuzzy

The Sugeno Fuzzy will be implemented to control enemy behavior, to choose the best behavior decision that will be executed based on player's state and enemy's state. A degree of membership will be calculated based on the fuzzy input which are distance, health and ammunition. The enemy's state then will be decided according to the input states.

According to Asus Naba, fuzzy logic is: "A calculations method with variables consists of words (linguistic variable) instead of numerical. Words used in fuzzy is not as precission as numerics, but it's closer for human" [5].

Sugeno Fuzzy is a branch in fuzzy algorithm. Fuzzy has a degree of membership in a range of 0 and 1. Fuzzy inferences in Fuzzy Sugeno divided in several parts [6] in the following:

1. Input fuzzification

   FIS takes inputs and determine its degree of membership in a fuzzy set.

2. Fuzzy logic operation

   The output for this operation is an antecedent degree of correctness implied by a single number.

3. Implication

   Is a process to determine a consequent or output from an IF-THEN rule according to the antacedent degree of correctness. This process use minimum value.

4. Defuzzyfication

   The output of defuzzyfication is a single number, this research uses averaging, by summing α-predicates from all rules multiplied by z-behavior.

The models of Sugeno Fuzzy are the following:

1. Zero-Orde Sugeno Model

   The zero-orde Sugeno Model is shown in the following rule:

   IF(x1 is A1) o (x2 is A2) o (x3 is A3) … o (xN is AN) THEN z = k

   Where Ai is the set of i-fuzzy as an anteseden, and k is a constant as a consequence.

2. One-Orde Sugeno Model

   The one-orde Sugeno Model is shown in the following rule:

   IF (x1 is A1) o … o (xN is AN) THEN z = p1 * x1 + … + pN * xN + q

   Where Ai is the set of i-fuzzy as an anteseden, and pi is a constant i. and q is also a constant of consequences.

2.2. Non-Playable Character

Autonomous character is a kind of autonomous agent used for computer animations and interactive media like video games and virtual reality. This agent represents a character in a story and has the ability to improvisize their behavior. This is the opposite form a character in an animated movie, in which their behavior has been written beforehead. In a game or virtual reality, the "avatar" are directed in real time, according to the game's state. In a video game, autonomous character are called Non-Playable Character. [7]

An NPC can be placed as many form, as an adversaries, provide assistance and guidance, to form part of a puzzle, to tell a story, to provide a backdrop to the action of the game, to be emotionally expressive, and so on [8]. In educational games for example, an NPC could be a character that "teach" the player about something and help them to understand how to play the game.

A game that features a rich interaction with NPCs can only benefit from AI that controls NPC with so many variations, expressive and able to mimics human like and believable behaviors [9].

The behaviour of autonomous character can be divide into several layers which are: action selection, steering, locomotion, as shown in Fig 1. In this research, the main concern are in action selection, which is the strategy of NPC's behavior.

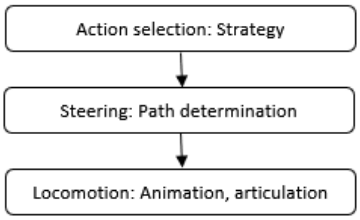


Fig. 1: Behavior Hierarchy

3. Methodology

3.1. Game Concept

This research will be implemented in "Healthy Hero", a 2D platformer game. The aim of this game is to introduce the importance of vaccination to children at age 6-12 years old. This game gives educational content for children about the danger of Measles and Rubela viruses [10], and also introduce the benefit of healthy food and the harm of junk food.

The setting for this game is inside the human body. The main character is a micro-patriotic hero that will fight against enemy which are Rubela viruses that infects human body. In this game, player will fight against those viruses and need to collect healthy food items to make sure the character has enough health and ammunition. The obstacles for each level are junk food items like pizza, soda, burger, and two kind of enemies, normal enemy and boss enemy.

Player can attack enemy using a weapon which is a syringe equipped with white blood cell as its ammunition, and vaccin as a shield. The normal enemy behavior controlled by randomizing state, and the boss enemy behavior controlled by fuzzy. The boss behavior are retreat, defense and attack, according to player and its own state. In depth explanation about enemy action are as follows:

1. Retreat

   Enemy will move in the opposite direction from the player, increasing distance from enemy to player.

2. Defense

   Enemy will blocks the player's attack by 50% less damage than normal.

3. Attack

   Enemy will shoot weapon towards player that will reduce player's health by 10%.

3.2. Game Design

In this part, we design the character, NPC, maps, items and obstacles for each level. The storyboard for this design are as follow:

Table 1: Storyboard

Main menu

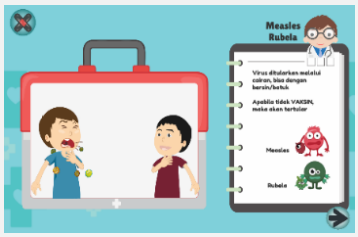
This main menu will appear after a splash screen. This menu contains play

button, knowledge button, setting, credit and exit.

Setting

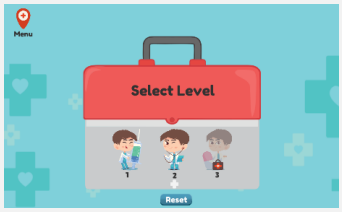
This menu is for sound effect and music setting.

Education Screen



In this screen, player will see a knowledge content about vaccine and healthy food.

Select Level



In this screen, player will be able to select between 3 levels. The next level will be locked until player finish the previous level.

Level 1 gameplay

Player will be faced against normal enemies and must reach score higher than 300 to proceed to the next level. Player need to collect healthy food items to increase health and avoid junkfood items.

Level 2 gameplay

In this level, there are normal enemies and one boss enemy. Player needs to reach score higher than 500 to proceed to the next level.

Level 3 gameplay

In this last level, normal enemies and two boss enemies must be defeated. Player must reach score higher than 700 to complete the game.

Game over screen

If the player's health is empty, then the game over screen will be showed, contains the total score the player obtained.

4. Implementation

In this research, we use Sugeno Fuzzy zero-orde because the outputs are a fixed constants, therefore can represents the behaviour correctly and can be applied to a game that need fast and real-time decision making [11]. The implementation of this methodology are shown in the following Fig 5.

Several phase will be conducted as follows:

4.1. Fuzzyfication

In this phase, the crisp value will be mapped in each fuzzy set as shown in Fig 2, Fig 3 and Fig 4. Three linguistic variable is used: Distance, Health and Ammunition.
Distance variable is the distance between main character and the enemy, ranged from 0 until 100, measured with in-engine API in Unity Engine [12].

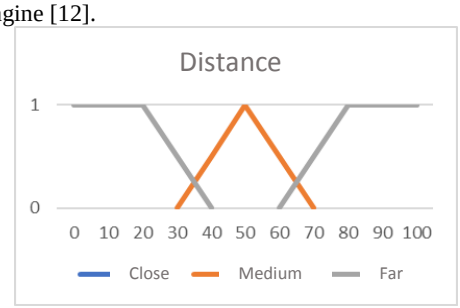


Fig. 2: Distance variable

Health variable is the enemy health, ranged from 0 to 100. Enemy health will be reduced by 10% if hit by main character's attack.

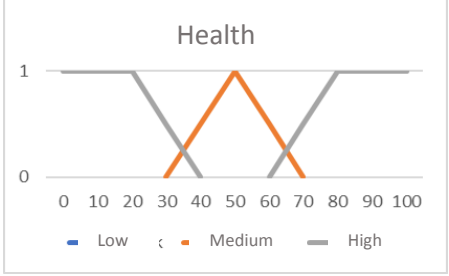


Fig. 3: Health variable

Ammunition variable is the player's weapon ammunitions ranged from 0 to 100. The ammunition will be reduced by 1 for each attact executed by player. It is worth noting that player has the tendency to do rapid attack eventhough there are not a lot of enemy ahead. So the reduction of ammunitions of above 1 of each attack is very likely.

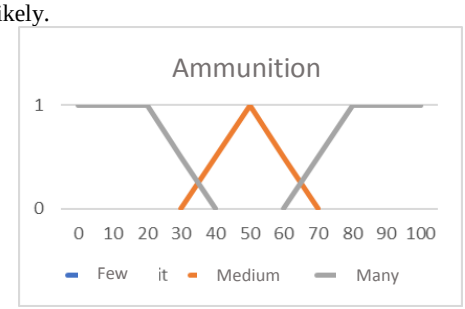


Fig. 4: Ammunition variable

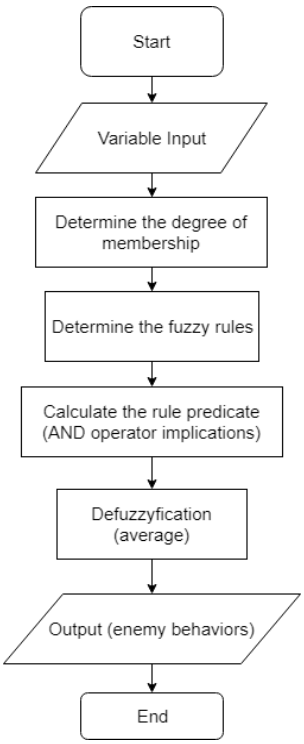


Fig. 5: Sugeno Fuzzy Implementation

The membership function used in this research are as follows:
Left shoulder trapezoidal curve

$$\mu[x] = \begin{cases} 1; x \leq 20 \\ \frac{40 - x}{40 - 20}; 20 \leq x \leq 40 \\ 0; x \geq 40 \end{cases} \quad (1)$$

Middle triangle curve

$$\mu[x] = \begin{cases} \frac{x - 30}{50 - 30}; 30 \leq x \leq 50 \\ \frac{70 - x}{70 - 50}; 50 \leq x \leq 70 \\ 0; x \leq 30 \; or \; x \geq 70 \end{cases} \quad (2)$$

Right shoulder trapezoidal curve

$$\mu[x] = \begin{cases} 1; x \geq 80 \\ \frac{x - 60}{80 - 60}; 60 \leq x \leq 80 \\ 0; x \leq 60 \end{cases} \quad (3)$$

4.2. Fuzzy Rules

There are 27 fuzzy rules which combine the three variables (distance, health, ammunition) as shown in Table 2.

Table 2: Fuzzy Rules

Num	Distance	Health	Ammunition	Action
1	Close	Low	Few	Defense
2			Medium	Retreat
3			Many	Retreat
4		Medium	Few	Attack
5			Medium	Defense
6			Many	Defense
7		High	Few	Attack
8			Medium	Attack
9			Many	Attack
10	Medium	Low	Few	Defense
11			Medium	Retreat
12			Many	Retreat
13		Medium	Few	Attack
14			Medium	Attack
15			Many	Defense
16		High	Few	Attack
17			Medium	Attack
18			Many	Attack
19	Far	Low	Few	Defense
20			Medium	Retreat
21			Many	Retreat
22		Medium	Few	Attack
23			Medium	Defense
24			Many	Defense
25		High	Few	Attack
26			Medium	Defense
27			Many	Defense

4.3. Fuzzy Inferences

At this phase, an implication function of MIN will be calculated to obtain the score of α for each rule. Based on the fuzzy rules the amount of α is 27. Each of α will be used to calculate the inference output.

$$\alpha = \min(\mu a, \mu b, \mu c) \quad (4)$$

Where α is degree of membership; μa = μ distance variable; μb = μ health variable; μc = μ ammunition variable.

1. Defuzzyfication

Defuzzyfication is where we obtain the output, as shown in the following formula:

$$z* = \frac{\sum a_i \, z_i}{\sum a_i} \quad (5)$$

Where z is a score for NPC action. The range of this score is between 1 and 3. The score of 1 implies the "retreat" action within the range of $x < 1$. The score of 2 implies the "Defense" action within the range of $1 < x < 2$. And the score of 3 implies the "attack" action within the range of $2 < x < 3$ or $x > 3$.

5. Results

5.1. Case Result

To evaluate our methodology, we take one case conditions, where the input variables are intendedly determined. Those variables are:

- Player to enemy distance : 64
- Enemy health : 62
- Player ammunition : 37

Based on the value of the input varibales, we can decide which rule that fulfill the condition, as shown in the following curves in Fig 6.

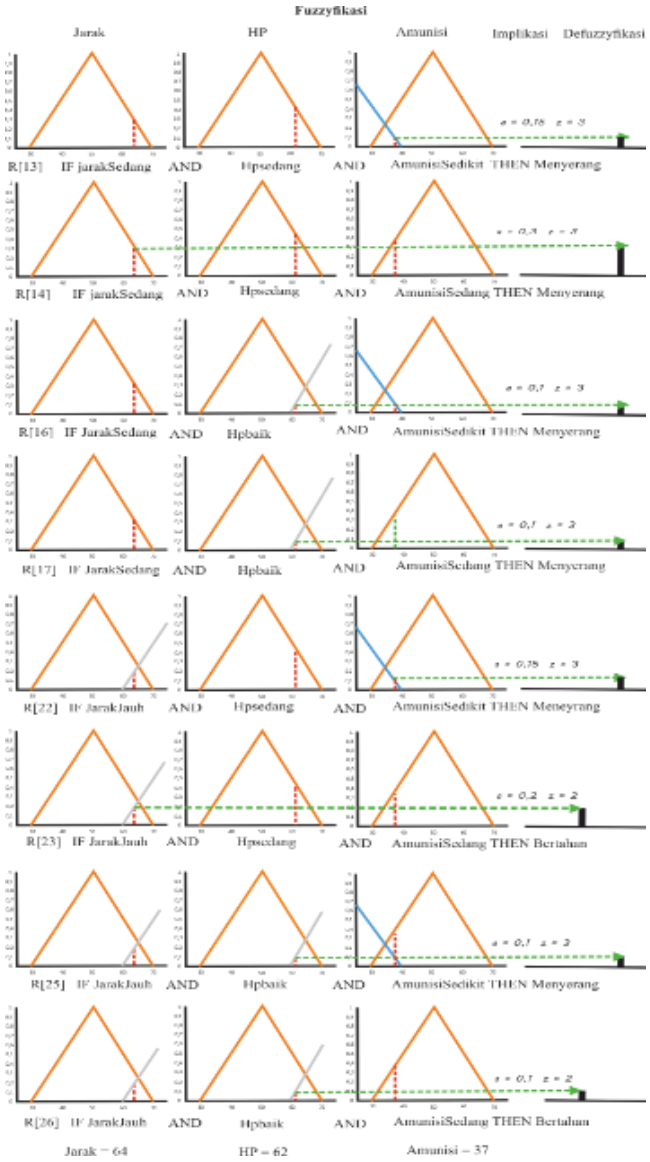


Fig. 6: Sugeno Fuzzy Calculation Curve

The result of defuzzyfication are as follows:

$$\frac{(0{,}15 * 3) + (0{,}3 * 3) + (0{,}1 * 3) + (0{,}1 * 3) + (0{,}15 * 3) + (0{,}2 * 2) + (0{,}1 * 3) + (0{,}1 * 2)}{0{,}15 + 0{,}3 + 0{,}1 + 0{,}1 + 0{,}15 + 0{,}2 + 0{,}1 + 0{,}1}$$

= 2.75 (Attack)

The result of defuzzyfication is calculated using average method, by summing all α-predicate multiplied with z-action for each rule then divided by the amount of α-predicate.

5.1. Game Implementation Result

The methodology implemented in the gameplay of the "Healthy Hero" game developed with Unity Game Engine. In this evaluation, the results of Sugeno Fuzzy in inferencing the behavior of the enemy are shown by the in-engine inspector window, as shown in Table 3, 4, 5.

By inspecting the inspector window for each scene, we can conclude the action performed by the boss enemy at **Defuzzyfication** part, in the **Output** textbox. This output is the result of fuzzy inference from the input variables, in the **Input** part. The recorded outputs are consistent with the pre-determined fuzzy rules shown in Table 2.

Table 3: Enemy with Retreat Action Behavior

Scene	Inspector	
	**Input**	
	Jarak	14
	HP	30
	Amunisi	90
	Fuzzifikasi	
	Jarak Dekat	1
	Jarak Sedang	0
	Jarak Jauh	0
	HP Buruk	0.5
	HP Sedang	0
	HP Baik	0
	Amunisi Sedikit	0
	Amunisi Sedang	0
	Amunisi Banyak	1
	Implikasi	
	Rule	
	Defuzzyfikasi	
	Sigma A	0.5
	Sigma AZ	0.5
	Defuzzy	1
	Output	kabur

Table 3 shows that with input variable of Distance (14), Health (30), Ammunition (90) are the conditions of Distance = close, Health = low, Ammunition = many, the defuzzyfication results is 1, which is Retreat.

Table 4: Enemy with Defense Action Behavior

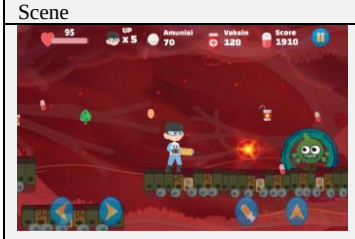
Scene	Inspector	
	**Input**	
	Jarak	8
	HP	60
	Amunisi	70
	Fuzzifikasi	
	Jarak Dekat	1
	Jarak Sedang	0
	Jarak Jauh	0
	HP Buruk	0
	HP Sedang	0.5
	HP Baik	0
	Amunisi Sedikit	0
	Amunisi Sedang	0
	Amunisi Banyak	0.5
	Implikasi	
	Rule	
	Defuzzyfikasi	
	Sigma A	0.5
	Sigma AZ	1
	Defuzzy	2
	Output	bertahan

Table 5: Enemy with Attack Action Behavior

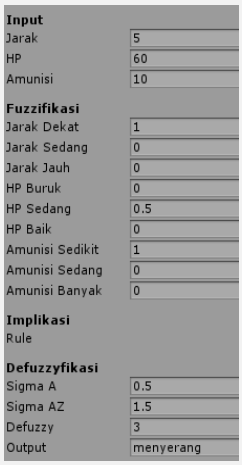
Scene	Inspector
	

Table 4 shows that with input variable of Distance (8), Health (60), Ammunition (70) are the conditions of Distance = close, Health = high, Ammunition = high, the defuzzyfication results is 2, which is Defense.

Table 5 shows that with input variable of Distance (5), Health (60), Ammunition (10) are the conditions of Distance = close, Health = high, Ammunition = low, the defuzzyfication results is 3, which is Attack.

5.2. Different Input Variables Result

We conduct several case evaluation using different input variable value to inspect whether our methodology is performing according to the ruleset that has determined before. The following Table 6 is the results of our evaluation.

Table 6: Sugeno Fuzzy Implementation Results

Num	Input			System Calculation		Manual Calculation	
	Dst	HP	Ammo	z	Action	z	Action
1.	21	70	90	3	Attack	3	Attack
2.	15	10	70	1	Retreat	1	Retreat
3.	35	20	60	1	Retreat	1	Retreat
4.	20	40	60	2	Defense	2	Defense
5.	4	40	30	3	Attack	3	Attack
6.	13	30	20	2	Defense	2	Defense
7.	25	30	80	1	Retreat	1	Retreat
8.	11	70	50	3	Attack	3	Attack
9.	6	60	40	2	Defense	2	Defense
10.	5	60	20	3	Attack	3	Attack
Error = $\frac{\text{Jumlah yang berbeda berdasarkan perilaku}}{\text{total ujicoba}} x\ 100\%$							0%
Success = 100% - Error							100%

By comparing the results of system calculations and manual calculations, the evaluation shows that the implementation for the game resulting in 100% correct NPC behavior according to the predetermined rules and player and enemy's state.

6. Conclusion

This research implements Sugeno Fuzzy in the gameplay of 2D platformer game "Healthy Hero" to determine the action of boss enemy NPC according to player's state and enemy's state. Testing evaluation shows that enemy action reacts correctly according to pre-determined fuzzy rules.

The testing using several different inputs variables whic are distance, health and ammunition resulting in 0% error and 100% correct. In this case, Sugeno Fuzzy proved can give artificial intellegence for the NPC and increase variations of NPC's behavior.

Some aspect can be re-tune to improve responsiveness of the enemy for example, tuning the range of distance so enemy can start to take action at a farther distance. But keep in mind that too far the minimum distance be set, the more unnatural behavior of enemy will occurs.

Further work can be done to increase the variations of NPC behavior, such as adding more boss type or adding more actions in each boss enemy. Adding more rules and applying Neural Network methodology to determine the enemy action can greatly improve the unpredictness and lifelike of NPC behavior.

References

[1] Yannakakis G., Togelius J. (2018) Artificial Intelligence and Games, Springer.
[2] Pirovano M. (2012) The use of Fuzzy Logic for Artificial Intelligence in Games. Technical report, University of Milano, Milano.
[3] Yunifa M.A., Ady W., Fachrul K., (2012) Pergantian Senjata NPC pada Game FPS Menggunakan Fuzzy Sugeno. Seminar Nasional Competitive Advantage.
[4] Purba, K., Hasanah, R., & Muslim, M. (2013). Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG.*Jurnal EECCIS, 7*(1), pp.15-20.
[5] Naba, Agus (2009), Belajar Cepat Fuzzy Logic Menggunakan MATLAB, Yogyakarta, Andi Offset.
[6] Kusumadewi, Sri, Sri Hartati. (2010). Neuro- Fuzzy Integrasi Sistem Fuzzy dan Jaringan Syaraf. Yogyakarta: Graha Ilmu.
[7] C. W. Reynolds (1999), Steering behaviors for autonomous charachters. In Proc. of Game Developers Conference, pages 763–782. Miller Freeman Game Group, San Francisco, CA.
[8] Treanor M, Zook A, Mirjam P. Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam Smith. (2015), AI-based game design patterns.
[9] Riedl, M., and Stern A. (2006), Believable agents and intelligent story adaptation for interactive storytelling. Technologies for Interactive Digital Storytelling and Entertainment, pages 1–12.
[10] Mitasari, Fajar, dkk (2017), Implementasi Logika Fuzzy pada Pembuatan Karakter Musuh untuk Game Single Fighter Berplatform Android. Jurnal SINUS Vol. 15 No. 1.
[11] Griffin D.E (2001) Measles Virus, in Fields Virology. Knipe DM , Howley PM, Griffin DE, Martin MA, Lamb RA, Roizman B, Straus SE, editors. 4th edition. Lippincott Williams & Wilkins.
[12] Unity Technologies (2018), Vector3.Distance. Unity Documentation, available online: https://docs.unity3d.com/ScriptReference/Vector3.Distance.html
[13] Abdiansyah, Primanita A. and Muliawan F. (2014) Fuzzy Logic Implementation on Enemy Speed Control to Raise Player Engagement. Proceeding of The 1st International Conference on Computer Science and Engineering, Vol. 1 No. 1.
[14] Hooshyar, D., Yousefi, M., & Lim, H. (2017). A systematic review of data-driven approaches in player modeling of educational games. Artificial Intelligence Review, 1-21.