IMPLEMENTATION OF PARTICLE SWARM OPTIMIZATION AND FINITE STATE MACHINE ALGORITHM FOR NPC GROUP BEHAVIOR AND DECISION MAKING IN MOBA GAMES

Mochammad Akbar Fadhlillah Mi'raj, Galih Hermawan

Universitas Komputer Indonesia
Jl Dipatiukur No. 112-114-116
E-mail : makbarfm@gmail.com, galih.hermawan@email.unikom.ac.id

ABSTRACT

Multiplayer Online Battle Arena (MOBA) is one of the game genre that originates from real time strategy types of game. The purpose of this game is for the player to destroy the other team base, before enemy team destroy ours. The player will control a character that can attack enemies. Player will get help from the group of NPC that called creeps. This creeps will comes in the game everytime. They will attack other team player, NPC and other team tower. Both teams will get NPC for helping them reach the base. Every player will meet them and battle them for purpose of leveling up their character, and make it more stronger. With that NPC cant become a feeder for the player. There must some sort of fighting back for the NPC to the player to make it more interesting gameplay/ NPC will move in a group consist of three creeps. This group must attack the player cooperatively. By using particle swarm optimization algorithm, group of NPC can move in a group and can attack the player.

This method will choose on of the NPC that have best position that is nearest to the player and the other NPC will follow that designated leader, and then they will attack the player or chasing the player. Not only by moving a group. NPC must take a decision when they do something. For this writer used finite state machine. The finite state machine can make the NPC to take the decision based on even that surround them. Even when NPC attack a tower and a player attack them, the NPC will take the decision and attack the player even when they still attack tower.

This will make the game challenging for the player and not too easy for the player. When the testing is done, it comes out as result that everytime the nearest NPC to the player is get choosen as the leader of the group and attack the player. And with FSM it comes out in a result that it can make the NPC to change the decision they making, but it still needs improvement when the NPC have three targets in front of them.

Keywords : finite state machine, particle swarm optimization, artificial intelligence, making decision, group activity

1. INTRODUCTION

Multiplayer Online Battle Arena (MOBA) is a type of video game where one or more players will play a certain character type where the task is to destroy the tower and the building structure on the opponent team. The game will be divided into two teams. Each team consists of 5 players and groups of NPC. This team will attack each other in an arena in which there are towers in each lane. MOBA game is multyplater video game, where there are one or more players playing in this game. The game will end if the main building of the opponent or player are destroyed. To achieve the goal the player will be assisted by a group of NPC (Non Playable Character) . Those NPC are called creeps. This NPC will move through the existing path. This NPC will meet opposing players and NPCs opponents in game. NPC also tasked to destroy the tower . Each of these creeps groups will fight players to discourage players so players can not easily complete its mission of destroying enemy headquarters.

In attacking NPC players must move in groups. NPCs can not separate when attacking players, players can exploit it and make NPC easier to beat by players. If NPCs are separated from other NPCs within the group it will also make the NPC unable to perform an optimal attack on the player. The NPC should also be able to determine what the NPC should attack. NPCs that only attack the tower and do not attack other than it will make the game take place easily. Because the NPC becomes unable to respond when attacked by another team. The NPC

should always attack the players to keep the game running challenging for the player .

On previous research by Victor do Nascimento Silva and Luiz Chaimowicz [1] said if the NPC in MOBA game is very influential in the course of the game. Creeps on MOBA is useful for player characters to gain experience points that are used for player character to get stronger. In this type of MOBA game artificial intelligence on NPC is used in how they move in groups and how they attack and determine what to be attacked by the NPC. In move NPC can not move in each. The NPC will continue to move in groups. From where they appear to the destination of the NPC. In decision-making NPC must also be able to determine which they must first select in action. Any action taken must have a direct effect on the player who will fight the NPC. So the problem that arises is the NPC must be able to move in groups and determine the decisions taken when in the game .

Algorithm Particle Swarm Optimization (PSO) is an algorithm that can be used to determine the behavior of the NPCs in groups. PSO algorithm is an algorithm that is inspired from the flock of birds that move together and aligned in the group. In the NPC group will be appointed one leader who will determine the behavior to be carried out the NPC group. Each NPC is initialized into a particle shape. The NPC will move in a direction corresponding to the direction of the converging harmony [2] . Later each NPC will convert it into a possible position when attacking an enemy tower , player or NPC.

Finite State Machines (FSM) is a method consisting of a series of State that can determine decision making. Each State may move from another State if it meets a specified condition. NPC can take action if get input in this case if meeting with player. In a study by Michael Waltham and Deshen Moodley called if FSM is the least used resources , FSM is suitable for use in games where NPC is not the core of the game terseb ut.

In this study the PSO algorithm will be used in NPCs that group as artificial intelligence for NPC and FSM group behavior as decision makers and will measure the performance of the algorithm.

1.1 Video Game

Video games is one of the type of entertainment that can be done actively. Watching movies, listening to songs is a passive entertainment activity. While playing video games is an activity where a user plays something that happens in the game. There is a world in the game that has rules that the user should understand. The rule is something that must be done so that the goal of the game is achieved.

1.2 MOBA Game

MOBA is a modification of the RTS game. If in the RTS game the player has to build many units and control it, in the MOBA game we only control one character and the troop unit is controlled by artificial intelligence. MOBA also has an RPG element where characters that players play can get stronger with level ups and have more ability at each level . .

1.3 Kecerdasan Buatan

Artificial intelligence is one part of computer science that makes computer machines for data to perform activities or jobs as and as well as humans do. Early creation, the computer is just a calculator only. Increasingly human, computer is currently used in many fields. Computers even claimed increasingly able to do human work. [3]

Humans are able to complete their work because they have intelligence in their brain. This intelligence is acquired by a process called learning. Humans are increasingly able to complete their work because through the process of learning is not for a moment, even with experiments many times, so that humans have the intelligence in completing the work and the problem.

In order for the computer to do the work as well as or better than humans, it takes an intelligence that must be planted in the computer. Artificial intelligence will provide methods for the computer so that computers can have reason like humans and make computers become more intelligent.

1.4 Finite State Machine

Finite State Machine is a methodology of control system design that describes the behavior and working principles of the system using the following three things: State , event and action . In a certain period of time the system resides on one State system that is being run. Then the system will switch or change to another State if it gets input in certain events, either from the outside or the components of the system itself. Changes in this state are also usually accompanied by actions that will be done by the system when responding to inputs that occur. The following actions can be either a simple action or a complex process of action. [6].

.

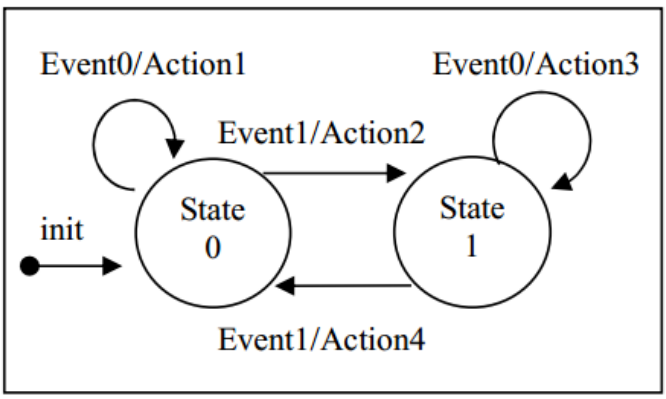


Figure 1. Finite State Machine [7]

Picture 4 shows the FSM with two State and two inputs and empah fruit outpu different looks in the picture, when it starts up, the system will transition to State (), at this point the system will bring up results Action1 case input an event () , whereas if event1 happens then action2 will be executed then system will transition to State 1 and so on.

Implementation of FSM can be applied to object-oriented programming or often called OOP. The advantages of OOP over others in the FSM implementation are its high flexibility and easy maintenance both on a system at any level. In addition, FSM benefits from the advantages of OOP in the reuse of code that has been typed. [6]

1.5. Particle Swarm Optimization

The PSO algorithm is an algorithm developed by Kennedy and R. Eberhart in 1995 that was inspired by the behavior of animal group movements such as fish, herbivorous animals, birds and insects such as ants and bees. Every object of the animal will be simplified into easily movable particles. This algorithm applies the properties of each individual in a large group and then combines these traits in order to solve the existing problem [2]. PSO algorithm is a method of artificial intelligence swarm intelligence based on collective behavior in a group and each individual in the group can self-regulate. In swarm intelligence the PSO algorithm is a method that can be used to set the best particle position. The concept of optimizing individual particles is introduced by Kennedy and R. Ebhert each particle has a fitness value evaluated by optimized functions, in addition the parameters of each particle have the following points of density constants, position coordinates, best position coordinates, best position coordinates globally. Of the parameters used above can form a position change that is done simultaneously in each particle. Before changing the position, the particle first changes its speed. This speed change uses the basic PSO equation [2] .

he particle update can be used with the PSO basic equation:: [5]

$$Vi(t+1) = Vi(t) + C1\ r1\ (\ Pbest\ i - Xi(t)\ ) + C2\ r2(\ Gbesti - Xi\ (t)\ ) \quad (2.1)$$

he particle position update can be done using the PSO basic equation: [5]

$$Xi(t+1) = Xi(t) + Vi(t+1) \quad (2.2)$$

Mencari nilai fitness

$$F\ (X) = \sqrt{((player.\ x — NPC.\ x)^2 + (player.\ y — NPC.\ y)^2)} \quad (2.3)$$

Di mana :

Vi (t) : particle velocity to (i) on iteration to (t)

Xi (t) :The position of the particles to (i) on the iteration to (t)

V : Velocity particle

X : Particle Coordinat

i : Particle number of (1,2,3N)

C1 dan C2 : The cognitive and social factor of this value is the value of the constant worth between (0 and 1)

r1 dan r2 : random number beetweemn 0 - 1

Pbest : Particle best position

Gbest : the position of particles that have the best position of other particles.

2. RESEARCH

2.1. PSO Analysis

This method of algorithm will determine the position of NPC when they chasing the player in the game. One NPC that have good position will be the leader of the pack and will leading the chase for the enemy. And the other will follow it. This is how to determine NPC that will be the leader :

2.1.1. PSO Case

As for solving the problem in the PSO required the following steps: :

1. Particle Inisialitation
   The initial step in which the number and coordinates of the particle will be determined first.
2. Calculate the value of fitnees of each particle
   To obtain the fitness value of each particle requires the coordinate value to be evaluated in the fitness function.
3. Determine the Pbest
   Pbest is the best position of every particle. It is the position occupied by the current particles.
4. Determine the Gbest
   Gbest value is the best particle value that has been through the fitness function and the most optimal.
5. Defining Velocity Value
   This value is derived from the velocity value equation to update the speed.
6. Calculate the value of new coordinates
   By entering the value of V into the new coordinate equation the new coordinate value will be found.

2.1.2. Input Analysis

In the PSO algorithm to reach a destination point each particle determines which particles have the best position first and then other particles that do not have the best position will renew the speed and position to achieve the position of the particles that have the best position to update the speed and position of the each particle is needed input, including:

1. The coordinate value of the position of each particle
   This value is obtained from coordinat of each NPC that have been determined first.
2. Fitness Value
   Fitness value are obtained from evaluating each coordinat value of the NPC into fitness function where that function is optimization

function.

3. Costanta Value Ci and C_2
   These value are constanta factor cognitive and social. C1 and C2 value are in between one into zero. For this research the value of the C1 and C2 is one..
4. Random Value r_1 and r_2
   This value is random number from zero to one. And will be implemented into the function value.
5. Pbest Value
   Pbest Value or personal best is value from the best position that are NPC having and will be taken from the NPC coordinat point.
6. Gbest Value
   Gbest value or global best is value from NPC position that have best position other than other NPC. Gbest value is get from NPC coordinat that have the best position. NPC that have the best Gbest is the one that have optimal fitness value optimum. Thes fitness value is depend on the function maximum or minimum..

2.1.3. Case Analysis

To be known :

Three NPCs are in the game arena. This NPC is at the beginning of the game. The player's character is on the other side of the game arena as well. NPCs and players have a coordinate point value that determines the point of location in the game. The NPC will move towards the player by following one of the NPCs that will be the leader. So the calculation of this case is as follows.

Case minimization

The fitness function equation

F (X) = √ ((player. x— NPC. x)^2+ (player. y— NPC. y)^2

Coordinate point player = (4 00, 2 80)

C1 = C2 = 1 // The values of C1 and C2 are determined (values between 0 and 1)

r1, r 2 = random value // values ranging from 0 to 1

Many particles = 3

Coordinate point value of initial NPC:

Coordinate NPC1: X1 = (1 70, 1 75)

Coordinate NPC2: X2 = (3 30, 1 90)

Coordinate NPC 3: X3 = (2 60, 1 50)

Velocity value of each particle at this time

Velocity NPC 1: V1 = 0

Velocity NPC 2: V2 = 0

Velocity NPC 3: V3 = 0

Resolution:

Iteration 0

Random value: r1 = 0, 1 r2 = 0, 4

The first step

Current NPC coordinate values::

Coordinat NPC1 : X1= (170,175)

Coordinat NPC2 : X2 = (230,190)

Coordinat NPC3 : X3 = (260,150)

Second step : Inputing coordinat value NPC into fitness function

F (X) = √ ((player. x— NPC. x)^2+ (player. y— NPC. y)^2

Fitness NPC1 : F(X) 1 = √ (400-170)^2+ (280-175)^2 = 152

Fitness NPC2 : F(X) 2 = √ (400-330)^2 +(280-190))^2 = 14

Fitness NPC3 : F(X) 3 = √ (400-260)^2 + (280-150)^2 = 91

Third Step

Pbest = coordinat value right now

Pbest NPC 1 : Pbest1 = (170,175)

Pbest NPC 2 : Pbest2= (330,190)

Pbest NPC 3 : Pbest3 = (260,150)

Fourth Step

The Gbest value is the value of the NPC coordinate point that has the most optimal fitness value which in this case will minimize the distance between the NPC and the player. The NPC with the fewest distance is the most optimal NPC and the leader. The fitness value in can:

Fitness NPC1 : F(X) 1 = 152

Fitness NPC2 : F(X) 2 = 14

Fitness NPC3 : F(X) 3 = 91

By seeing the fitness value of the NPC, NPC that has lower value than the other is NPC number 2 than the Gbest value is NPC number 2 (330,190).

Fifth Step

Other NPC that is not the leader are updating their velocity with this function:

Vi (t+1) = Vi(t)+C1.r1(Pbest i – Xi(t)) + C2.r2(Gbest i –Xi(t))

Updating velocity of every each NPC :

NPC1 : Vx1 (1) = 0 + 0.1(170– 170) + 0.4(330 – 170) = 80

Vy1 (1) = 0 + 0.1(175-175) + 0.4(190-175) = 7.5

NPC3 : Vx3 (1) = 0 + 0.1(360- 360) + 0.4(430 - 360) = 35

Vy3 (1) = 0 + 0.1(150-150) + 0.4(190-150) = 20

Sixth Step

Other NPC that not the leader update their position with new velocity with this function :

Xi (t+1) = Xi (t) +Vi (t+1)

Updating velocity each NPC :

NPC 1 : X1(1) = 170 + 80 = 150

Y1(1) = 175 + 7.5 = 182.5

NPC3 : X3(1) = 260 + 35 = 295

Y3(1) = 150 +20 =170

New coordinat for each NPC

X1(0) = (170,175)

X3(0) = (260,150)

Coordinat NPC after updating position

X1(1) = (150,182.5)

X3(1) = (295,170)

Sevent Step

Checking position of the NPC if the position is convergen or not. With seeing the result of the iteraion 0 upthere. If not than repeat the iteration.

2.2.4. Updating Position After NPC

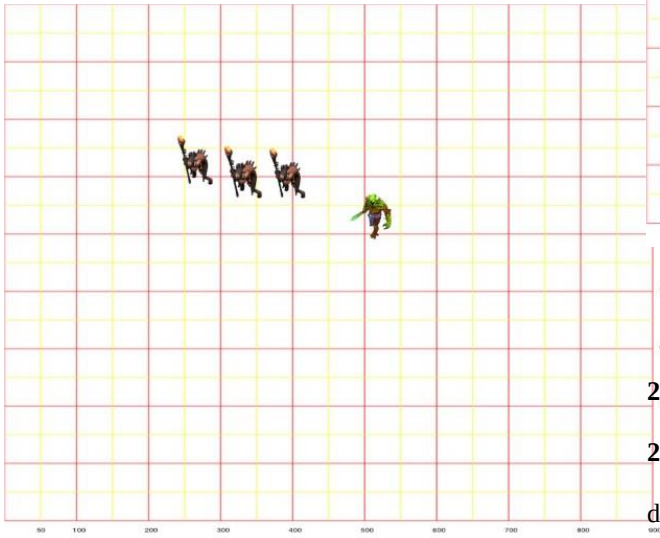


Figure 2. Starting Position

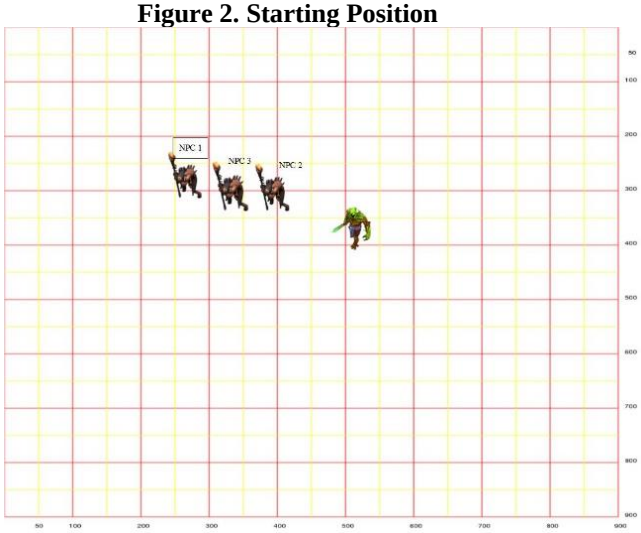


Figure 3. First Iteration

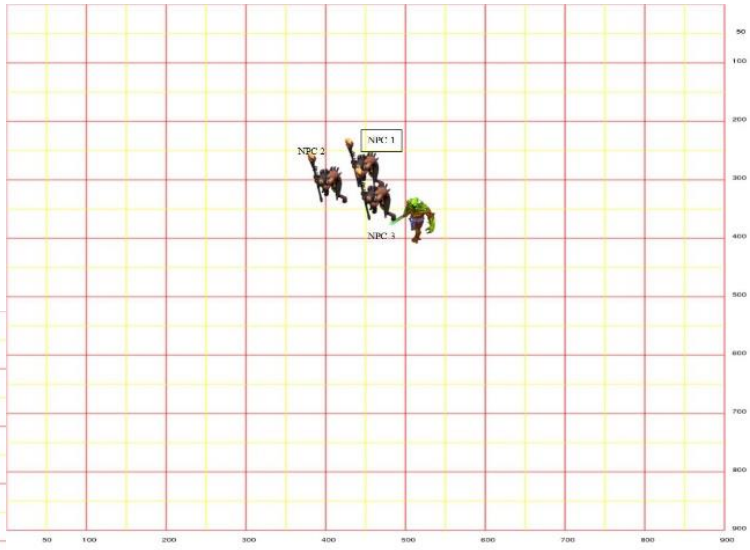


Figure 4. Second Iteration

2.2. FSM Analiysis

2.2.1. Finite State Machine pada NPC

In this research FSM will be used NPC for decision making in this game. State that is active at one time can only active one only. Each state will represent an action such as attacking and selecting a target. The first state that was active at the beginning of the game was to find the tower in the path already provided.

The first state when the game starts is to reach the first tower in this case the NPC will search the path with the path finding algorithm . The initial state of search for this tower will continue to be active until the NPC meets the tower. If you have found the tower then the NPC state will transition to destroy the tower. The state will continue to be active until the tower is destroyed. If the state meets the enemy NPC then the state will transition towards destroying the enemy. The NPC will execute only one state at a time. Gambara initial state can be seen in the figure below:

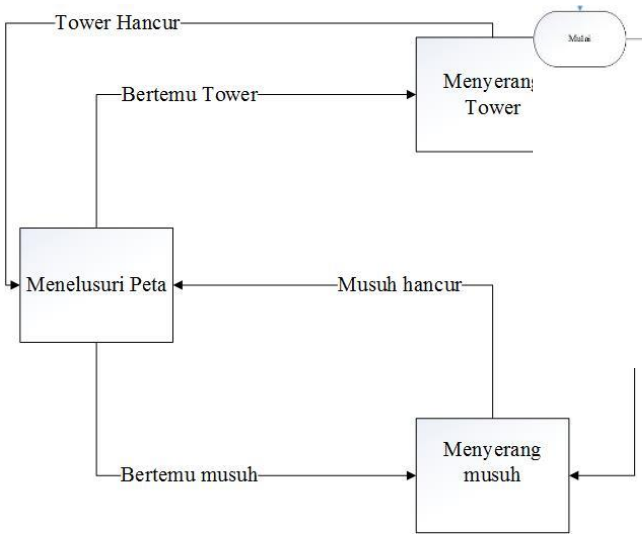


Figure 5. NPC State

2.2.2. Attacking Tower State

This state is a direct transition from the state looking tower when the NPC has found the tower . NPC is done first is to check whether near the tower there are NPC enemy or opponent. If there is an enemy, then the NPC will run the state attacked the enemy first. This state will transition back into a state attacking the tower when the enemy is no longer within NPC range. this state will make the NPC attack tower until the tower is destroyed. the state will transition back to the tower if the tower is destroyed and there are no enemies around the NPC.

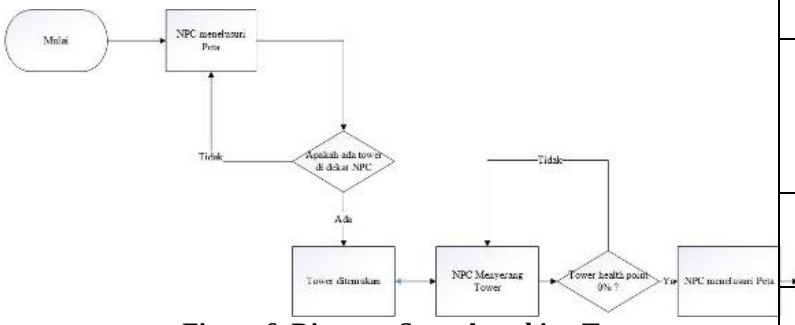


Figure 6. Diagram State Attacking Tower

2.2.3. Attacking Player State

This state will be done when the NPC is in close proximity to the player. When chasing NPC players will use the PSO method when attacking in groups. The NPC will check things such as distance with players, and enemy health points . When the NPC has checked it then the NPC can attack the enemy. NPCs can attack in groups or individuals. After the enemy's health point reaches 0% then the NPC will transition to another state . The NPC will prioritize attacking player characters first rather than NPC characters.

Figure 7. Diagram State Attacking Player

2.5 System Testing

Testing method on MOBA game for PSO and FSM method is test of decision making accuracy for NPC when getting input from outside.

2.5.2. FSM Testing

In testing the finite state machine NPC will be stored in a condition and will be seen how the NPC will take a predetermined decision. For NPC state move in game arena, NPC will be set in group and will try to be given condition. In this state the PSO algorithm will be used.

Table 1. Testing FSM

Result Testing			
Condition	Ecpected Result	Observation Result	Conclusion
NPC moving into tower area	NPC move to tower	NPC Move to tower	Success
NPC is near player	NPC chasing player	NPC chasing player	Success
NPC chasing player	One leader NPC followed by other NPC	One leader NPC followed by other NPC	Success

In testing for state attacking players, NPC and tower will be unified. NPC will basically attack targets that have a closer position to the NPC, but for testing this time, healthpoint will be a new parameter for testing and will see whether the NPC can make decisions correctly and choose the target that should be selected.

Table 2. State Attacking

Conclusion			
Condition	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan

NPC near enemy NPC	NPC attack Npc	NPC attack Npc	Success
NPC meeting player when attacking	NPC attacking player	NPC attacking player	Success

2.5.1. PSO Testing

Correcting test NPC leader election position is tests performed on the game to customize the optimal fitness value by minimizing the function of the NPC who was elected leader

$$F(X) = j\,(player.x - NPC.x)^2 + (player.y - NPC.y)^2$$

Where :
player.x is coordinat player X
player.y is coordinat player Y
NPC.x is coordinat player X
NPC.y is coordinat player Y
Player coordinat (400,280)

Table 3. First Testing

No	Result				
	Nama	Coordinat NPC		Koordinat Pematin	
		X	Y	X	Y
1	NPC 1	310	116	469	406.45
2	NPC 2	383	322.99		
3	NPC 3	437	152.87		

Those value will be inputing into the function and the result will be come

Tabel 4. Hasil Uji

Nomor	NPC	Hasil Pengujian	
		Nilai fitness	Keterangan
1	NPC 1	329	Follower
2	NPC 2	119	Leader
3	NPC 3	255	Follower

For the second testing :

Tabel 5. Testing Conditon Number 2

omor	result				
	Nama	Coordinat NPC			
		X	Y		Y
1	NPC 1	579	294	463	342
2	NPC 2	437	151		
3	NPC 3	381	196		

Nilai-nilai tersebut akan dimasukan ke dalam fungsi fitness untuk menentukan NPC yang paling cocok untuk dijadikan pemimpin.

Tabel 6. Hasil Uji kondisi Kedua

Nomor	NPC	Hasil Pengujian	
		Nilai fitness	Keterangan
1	NPC 1	125	Pemimpin
2	NPC 2	192	Pengikut
3	NPC 3	267	Pengikut

3. CONCLUSION

In this study has produced an artificial intelligence on the NPC in miniMOBA game , which is used as a group behavior and also as a decision maker . NPCs can move in groups and attack players in the game arena. NPCs can also make decisions when there is input from players or other NPCs as well.

The FSM algorithm used gives a result of 80% for NPC how they should act when a condition occurs. For testing to determine which leader can be on a test case, successfully establish a leader with a fitness score of 125.

The system that is currently built can still be developed again in accordance with the needs of users and also can develop a better gameplay . Suggestions for further development of the system that can be used per t counterweight namely:

a. Adding new lane in the arena gameplay

b. Adding new state for the NPC so the NPC can make varies new decision..
c. Adding new NPC so the player can have more challenge in the gameplay.
d. Adding new player in the gameplay.
e. Giving new map so the NPC must make decision to where they going..
f. Make new type and diffrent NPC.

BIBLIOGRAPHY

[1] V. do Nascimento Silva dan L. Chaimowicz, “On the Development of Intelligent Agents for MOBA Games,” Brazilian Symposium on Computer Games and Digital Entertainment, vol. 14, hal. 143-152, 2014.

[2] S. Mu'min, M. Hariadi dan S. M. Susiki Nugroho, “Pergerakan Otonom Pasukan Berbasis Algoritma Boids Menggunakan Metode Particle Swarm Optimization,” Journal of Animation and Games Studies, vol. 1, no. 1, hal. 1-16, 2015.

[3] S. Kusumadewi, Artificial Intelligence (Teknik dan Aplikasinya), Yogyakarta: Graha Ilmu, 2003.

[4] R. Roedavan, Unity Game Engine, Bandung: Informatika, 2016.

[5] R. E. dan Kennedy, “Particle Swarm Optimization,” dalam Conference on Neural Networks, Perth, Australia, 1948.

[6] M. Fauzan Rahadian, A. Suyatno dan S. Maharani, “Penerapan Metode Finite State Machine pada Game "The Relationship",” Jurnal Informatika Mulawarman, vol. 11, no. 1, hal. 14-22, 2016.