Dynamic Evolution Behavior for Non-Player Character on Space Shooter Game Using NSGA-II

Darmawan Aditama[1], Rizqi Putri Nourma Budiarti[2], Supeno Mardi Susiki Nugroho[3], Mochamad Hariadi[4]
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
darmawan14@mhs.ee.its.ac.id1, rizqi.putri14@mhs.ee.its.ac.id2, mardi@ee.its.ac.id3, mochar@ee.its.ac.id4

Abstrak – ***In Space Game, the NPC enemies have a static ability or behaviour where the player can be easily learn to adapted and challenge the NPCs. In the end, it makes the player become boring and don't want to play the game anymore. This research is develop an intelligent NPC that can be adapt to player style in playing the game. The NPC can be evolve and adjust itself based on the player behaviour in playing game and giving a proper difficulty to the game itself. This implementation will be make game more fun and enjoyable. NSGA II is the algorithm that will used to arrange the evolution of NPCs. This algorithm is used to evaluate the parameter that will determine NPCs behaviour change. Beside that, this research also used NSGA-II to optimized the paramater before it used to determined the NPCs behaviour change in every level of the game. Simulation itself used NSGA-II to produce optimal solution in graphic representation based on two objective parameter, speed and health. Result of simulation shown the optimized result will be produce the best solution after the five generation with total population is 50. crossover probability (p_c)=1, mutation probability (p_m)=1/n, index of distribution crossover (η_c)=20, index of distribution mutation (η_m) =20.***

Keywords: ***Genetics Algorithm, Artificial Intelligent, Behavior Evolution***

1. INTRODUCTION

In many classic type of game, like shooter game, such as space shooter. The NPC of enemies usually only have a static behaviour. It makes players easily to adapt the game and the game is not challenging anymore.

This research is develop an intelligent NPC that can be adapt to player style in playing the game. The NPC can be evolve and adjust itself based on the player behaviour in playing game and giving a proper difficulty to the game itself.

The concept of a fairness distribution of the ability between different strategies or several player is the quality of the game [1]. The evolution of adaptive behavior to change the speed and health on the NPC is needed, in order to match the ability of the player. The game will be more interesting because the NPC can evolve towards the speed and health by seeing the behavior of the player.

To provide the behavioral evolution of NPC, we use Non-dominated Sorting Genetic Algorithm II (NSGA-II). NSGA-II is a genetic algorithm for multi-objective optimization based on non-domination. NSGA-II is used to determine the optimizations on the NPC's parameters such as speed and health.

2. IMPLEMENTATION METHOD

2.1 Game Design

FPS (First Person Shooter) is a type of shooting game, with a view of the character in the game that we are playing, usually from the point of view in the eyes showed only hands and arms. The FPS game picks the missions for a particular purpose. The characteristic of this game is the use of long-range weapons [2].

In the game Space Shooter, the NPC has the ability to follow the movement of the player and can be fired towards the player, as seen in Figure 1, 2 & 3.

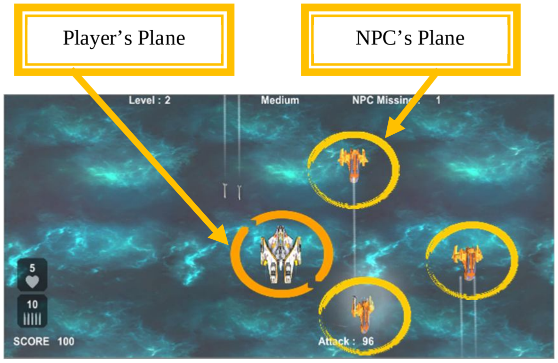


Figure 1. Space Shooter Game Plan

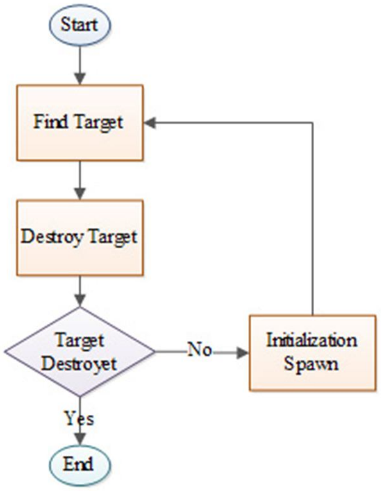


Figure 2. Flowchart of Space shooter Game

2.2 The Dynamic Evolution of the NPC Speed and Health

The game that cannot be changed adaptively is very tedious because the player will be easy or too difficult to face the NPC. Therefore, an adaptive ability in the game is needed, where NPCs has ability to be determined by itself. So the NPC is able to grow by itself by studying the ability of the player and the cycle of constant changes of the player.

Dynamic evolution of NPC that has the complex ability will be useful for the development of the pattern of the game that can be more interesting. The evolution of the speed and health that intended is the dynamic evolution of NPC when faced with the player, that can increase speed and optimal health by itself while facing the player.

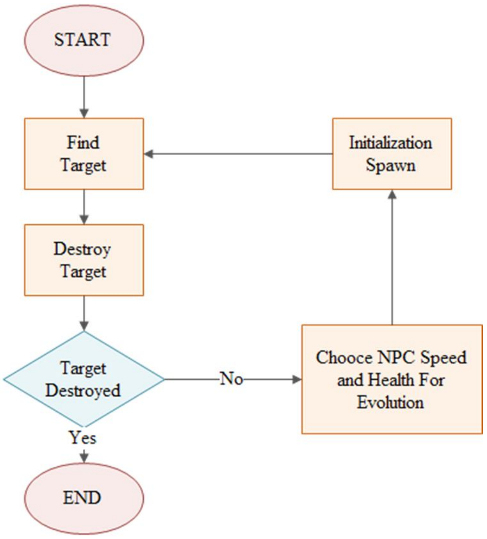


Figure 3. Flowchart of dynamic evolution of NPC

2.3 Multi-Objective Optimization

Optimization of Multi-Objective (also known as the programming of multi-purpose, optimization of vectors, optimization of multiple criteria, optimization of multi-attribute or optimization Pareto) is an area of several decision-making criteria. In here, the issues that related to mathematical optimization involving more than one objective function to be optimized simultaneously.

By minimizing the attack while maximizing protection of the troops in this research is an example of a multi-objective optimization problem that involving two or three objectives in each purpose. In practical problems, there can be more than three objectives [3].

2.4 NSGA II

The main step in the NSGA-II is the n offspring (St) is created from the n parent (Pt) using the standard of genetic algorithm. Overall the population (Wt) measuring 2n formed by merging St and Pt.

$$W_t = P_t \cup S_t \qquad (1)$$

Sorting Sorting non-domination applied by $W_{t,}$ then the new population will be loaded one at a time by differently non-domination solutions. Because the population of W_t is 2n, only half of these amount that will be the new population. After that, choose the non-domination solution of Pareto front with the high diversity and discard the remaining solution [2].

3. THE SCOPE OF RESEARCH

In a game, It's become very needed for a player to learn the game control, so it becomes important to improve the player's ability progressively in the game to maintain the flow conditions [4]. Flow become a popular technique for measuring the "fun" of a game, however measuring challenge and competence to provide flow is not an easy thing.

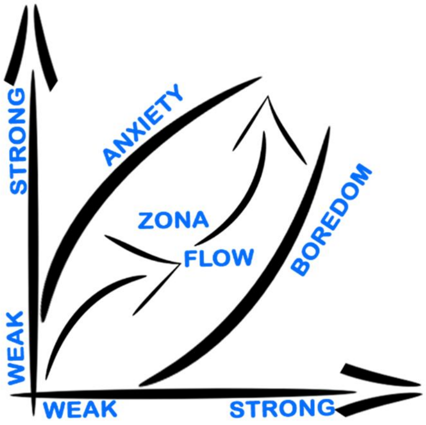


Figure 4. Flow Zone and Design Flow Adaptation Experiences

Figure 4 shows the flow diagram, which is the visual graph of the flow zone and design adaptation of the player experience through the choice of flow that is deliberately constructed by the players and becomes an experience for the player.

3.1 Gameplay

Space shooter is a game that aims to provide an active experience response to the behaviour of the NPC, in which the player must destroy as many NPC so the player can defeat the boss at each level to be able step up to the next level without running out of health. The basic concept of this game is to create an environment spaceship battles between players and NPC. Where both of these planes (NPC and the player) have been equipped with the weapons to destroy each other. In the player's plane there are two types of weapons. There are gun and missiles and also bonus items such as health, damage and an active shield if the player can pick up bonuses that dropped by NPC's plane while successfully destroyed by the player's plane. In the NPC's plane, there are four types of weapons such as a gun, missile, rocket, and suicide.

3.2 Scenario In Space Shooter Game

In the game space shooter that has existed, the election scenario on the level of difficulty in the game is tend to be static (easy, medium, and hard) which changes the type of speed and health of the enemy into static. So the player can easily guess the pattern of NPC's attacks because it is too easy to deal with or NPC can be very difficult while faced by the player when the player made a mistake when choosing the level of difficulty when playing the game. So it can give a tedious effect if during playing the game, the NPC is too easily defeated or too hard to defeat.

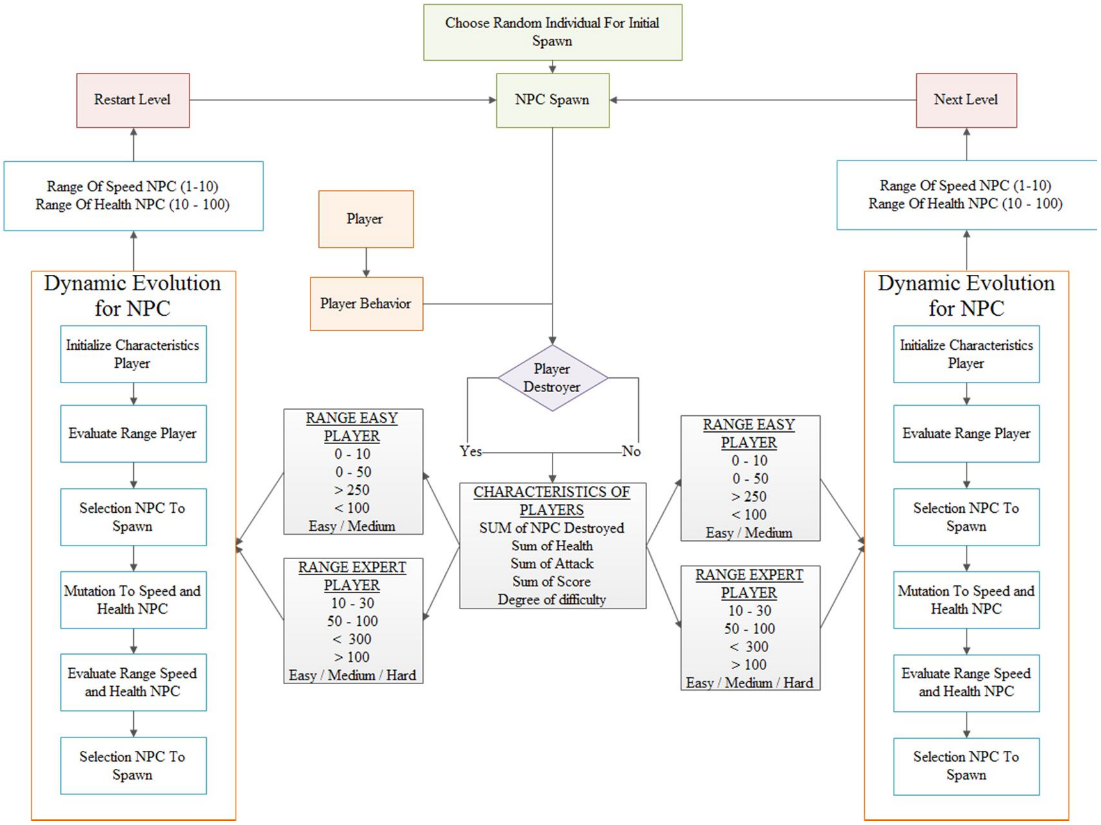
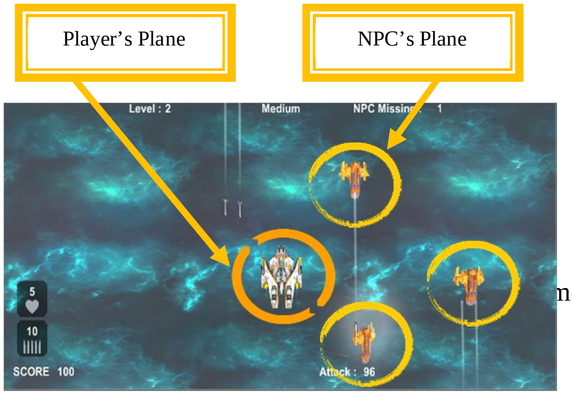


Figure 5. Scenario of Difficulty Level In The Static Game

3.4 Scenario Dynamic Evolution of NPC

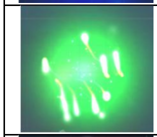
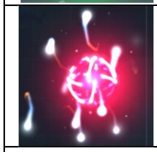
In this paper, the dynamic evolution scenario of NPC that can adapt to the behavior of players, the game scenario in this paper was designed as follows. Player is assigned to destroy as many enemies (NPC) to reach the final stage or meet the boss. If the player can destroy the boss, the player can continue the game at the next level. But if the player's plane is destroyed before defeating the boss, the player must repeat the game and started all over again.



me

In Figure 6 that shows the player is facing an enemy in the next level after the player can defeat the boss at the previous level, so the player can step up to the next level.

Table 1. Items and Player’s Weapons

Items	Fuction
	Items for Adding a Missile In Weapon Stocks
	Items for Adding The Player’s Health
	Items for Activating The Player’s Shield
	Gun Weapon that used by The Player to face the enemy
	Rocket Weapon that used by The player to face the enemy

During the game progresses the player's plane equipped with two types of weapons, there are the gun and missiles, as well as three kinds of items that can be obtained by the player if the player can destroy enemies and pick up items that fell. In Table 2 are described in detail, the specification of the player's plane. Starting from the names of items and weapons and also the function during the game.

Table 2. Specification of The Player’s Plane

Player’s Speed	Player’s Health	Items	Items Function	Type of Weapons	Player’s Damage Weapons
10	100	Health	+10	Gun	10
		Weapon	+1		
		Armor	10 s	Rudal	30

Not only the player but also the NPC. Because during the game, the NPC's plane also has many variants weapons despite not having the items to support the NPC's player while facing the player's plane as shown in Table 3.

Table 3. The Variations of NPC's Weapons

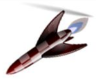
Weapons	Functions
	Gun Weapon that used by NPC to face the player
	Missile Weapon that used by NPC to face the player
	Rocket Weapon that used by NPC to face the player
	Suicide Weapon that used by NPC to face the player

Table 4. Specification of NPC's Plane

Speed Npc	Health Npc	Type of NPC's Weapon	Damage Weapon of NPC
10	100	Gun	10
		Missile	20
		Rocket	30
		Suicide	40

In Table 4 are shown the specification of NPC's plane in detail. Start of speed, health, the types of NPC's weapons that belonging to NPC and also the damage weapons that available in NPC.

There are 10 levels in this research trial, which at each level chromosome NPC will always change according to the player's behaviour.

However in the first level of the space shooter game, all four variants of the weapon would come out with speed and constant health that is still appropriate in Table 3. Due to the first level, is used as the test data on the behavior of the player, so that at the second level up to tenth level, the constant value of speed and health of the NPC can be change according to player behavior.

The purpose of the paper is to determine the dynamic evolution of the NPC at the next level by looking at the parameters of the players in the previous level. These parameters include player score, player heatlh, NPCmissed, and the player damage. These parameters are used to determine the dynamic evolution of the NPC at the next level.

During the game was played, the parameters of the NPC such as speed (f_1) and heatlh (f_2) will be changed according to parameters of the player. Because of these three genes used for the dynamic evolution of the NPC while facing the player.

If the player wins at the previous level with the condition of its heatlh in low level, gets a medium score, NPCmissed in high status and too much damage with an average value is less than 30, then in the next level, NPC will increase the level of difficulty R_3 which can be likened to the interval value in the dynamic evolution of the NPC (Rn) such that shown in table 5.

Table 5. The Interval value of dynamic evolution of the NPC

No	Dynamic Evolution NPC (R_n)
1	R_{10} : If $90 < (f_1\ f_2) \leq 100$
2	R_9 : If $80 < (f_1\ f_2) \leq 90$
3	R_8 : If $70 < (f_1\ f_2) \leq 80$
4	R_7 : If $60 < (f_1\ f_2) \leq 70$
5	R_6 : If $50 < (f_1\ f_2) \leq 60$
6	R_5 : If $40 < (f_1\ f_2) \leq 50$
7	R_4 : If $30 < (f_1\ f_2) \leq 40$

8	R_3 : If $20 < (f_1\ f_2) \leq 30$
9	R_2 : If $10 < (f_1\ f_2) \leq 20$
10	R_1 : If $0 < (f_1\ f_2) \leq 10$

Starting a beginning of the game state, the initial spawn NPC can be analogized as the default, as the player's ability level test on the first level in the game. After the first level is completed, then the existing parameters on the player (player score, player health, NPCmissed, player damage) will be calculated to determine the dynamic evolution of the NPC, where the NPC is raised at the next level using the weapon gun, missile, rocket, or suicide and evolutionary parameters on NPC (speed (f_1) and health (f_2)).

3.5 NSGA II Scenario In The Game

In the Space Shooter game, the research is focused on the optimization of the dynamic evolution of the NPC. Optimization of dynamic evolution of the NPC must pass through the various steps or stages, starting from making space shooter game including with the item, NPC's dynamic evolution scenario, multi-objective optimization of the dynamic evolution of the NPC.

The evolutionary scenario of the dynamic evolution of the NPC behavior using NSGA II in the Space Shooter game as shown in Figure 7.

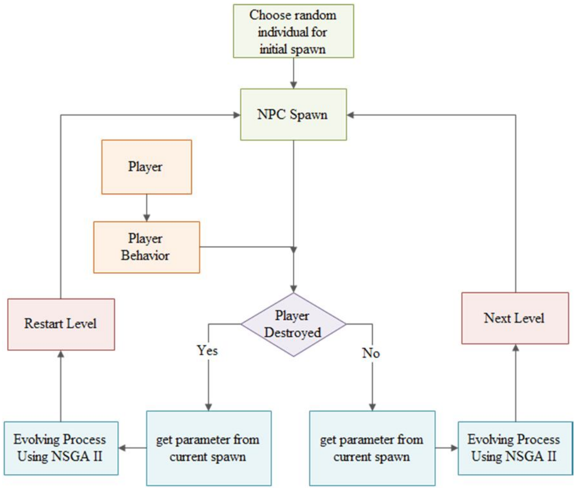

Choose random individual for initial spawn
NPC Spawn
Player
Player Behavior
Restart Level
Player Destroyed
Yes
No
Next Level
Evolving Process Using NSGA II
get parameter from current spawn
get parameter from current spawn
Evolving Process Using NSGA II

Figure 7. Diagram Block of The Game

3.6 Objective Function

Multi-objective optimization is an optimization problem that contains more than one objective function. Multi-objective optimization can be defined in an mathematical expression [5].

In this research, the function to be optimized is the adaptive behavior of the NPC to face the player behavior, where the goal is to be achieved by maximizing and minimizing the evolution of the NPC speed and health. In here, we attempting to the objective value of the player, so The objective function value of the player can be seen in Table 6.

Table 6. Objective Value of The Player

Speed Player	Health Player	Items	Items Function	Type of Player's weapon	Damage Player's weapon
10	100	Health	+10	Gun	10
		Weapon	+1		
		Armor	10 s	Rocket	30

However not only the objective function value of the player which is used to the dynamic evolution of the NPC. Because the objective function of the NPC is also necessary to find the point of maximum and minimum evolution NSGA II.

Table 7. Objective Value of The NPC

Speed Npc	Health Npc	Type of NPC’s weapon	Damage NPC’s weapon
10	100	Gun	10
		Missile	20
		Rocket	30
		Suicide	40

3.7 Objective Function

In here, the main purpose of the research study is contained at 3 fitness function as a parameter the dynamic evolution of the NPC, the fitness function is the speed (f_1) and heatlh (f_2).

The objective function value is needed to be able to change the fitness function corresponding to player's behaviours, so the objective function value of the player's parameters player score, player health, npcmissed, and also player damage that is used as a "point of evolution" for npc dynamic can evolve to the next level.

The parameter value such as score, npcmiss, heatlh and damage of the player is used as the parameter to provide multi-objective optimization value in the dynamic evolution of the NPC (changing composition of chromosomes and genes the NPC to face the player).

So the optimalization is really needed to increase the objective function at speed (f_1) and health (f_2) of NPC by distinguishing the parameter of player's objective function. Then it can be made the equation of the player's objective function as follow:

1) amount of the NPC

In simulation, the equation to get total value from all the spawn NPCs is also needed.

$$n = 30 + 1 \tag{2}$$

Where n is the sum of NPCs spawn in each level.

2) score of the player

Score is one of the parameters that used to facilitate changes of the fitness function in the dynamic evolution of the NPC. Where the objective value is taken from the parameter value to the player. The function to calculate the score as follows:

$$sp = sd * 10 \tag{3}$$

where :

sp = score of the player
sd = sum of NPC destroyed

3) health of the player

Health is one of the parameters that used to facilitate changes of the fitness function in the dynamic evolution of the NPC. Where the objective value is taken from the parameter value to the player. The function to calculate the heatlh as follows:
set initialization ha=100

$$hp = ha - \left(\left(\frac{0{,}03 * a * b}{1 + (0{,}3 * (a * b))} \right) * 100\% \right) \tag{4}$$

Where:

hp = health of the player
ha = Setting of health
a = the number of gun shots NPC's weapon to the player
b = damage NPC’s weapon to the player

4) Sum of NPCMissed

NPCMissed is one of the parameters that used to facilitate changes of the fitness in the dynamic evolution of the NPC's speed and health. Where the objective value is taken from the parameter value to the player. The function to calculate NPCMissed as follows:

$$\mathrm{NM} = (\mathrm{ss} - \mathrm{sd}) / \mathrm{ss} \tag{5}$$

Where:

NM = Sum of NPCMissed
sd = sum of npc destroyed
ss = sum of npc spawn

5) Damage Player On NPC

Damage is one of the parameters that used to facilitate changes of the fitness function in the dynamic evolution of the NPC. Where the objective value is taken from the parameter value to the player. The function to calculate the damage as follows:

$$\mathrm{dp} = (\mathrm{c} * \mathrm{d}) - \left(\left(\frac{0{,}3 * \mathrm{a} * \mathrm{b}}{1 + (0{,}03(\mathrm{a} * \mathrm{b}))}\right) * 100\%\right) \tag{6}$$

Where:

dp = player damage
a = sum of gun shot NPC's weapon to the player
b = damage NPC's weapon to the player
c = sum of gun shot the player's weapon to the NPC
d = damage player's weapon to the NPC

When the objective function of the player's formula is made, then this formula will be used to optimize the objective function in the dynamic evolution of NPC. However, the aim of this research is to create the dynamic evolution from speed (f_1) and health (f_2) of the NPC, to counterbalance the player's behavior when playing the game. Because of that, the objective function's formula is made to determine the dynamic evolution for speed (f_1) and health (f_2) of the NPC as follows:

1) Speed of the NPC (f_1)

Speed is the parameter that used to improve the movement of the NPC. Speed is one of the fitness function that becomes one of the goals in the NSGA-II. Speed is symbolized by f_1. Objective function value of parameters player such as player score (sp), player health (hp), NPCmissed (NM), and the player damage (dp) is used to control the appropriate changes of the speed parameter in the game which can then be referred as a speed evolution (se).

$$f_1 = \left(\frac{(150 * x_1 + 0.5 * x_4)}{2}\right) * \frac{\mathrm{gs}}{\mathrm{w}} * \left(\frac{x_1 * x_4}{2}\right) * \mathrm{n} * \mathrm{se} + \left(\frac{\mathrm{hp} * \mathrm{dp}}{2}\right) \tag{7}$$

Where:

n = sizeNPC
gs = speed
se = speed evolution

$$\mathrm{w} = x_1 + x_2 + x_3 + x_4 \tag{8}$$

Where:

w = weight
x_1 = genScore
x_2 = genHealth
x_3 = genNpcMissed
x_4 = genDamage

2) Heatlh of the NPC (f_2)

To add a number of health from the NPC. Health is one of the fitness function becomes one of the goals in the NSGA-II. Health symbolized by f_2. Objective function value of parameters player such as player score (sp), player health (hp), NPCmissed (NM), and also player damage (dp) that is used to control the appropriate changes of the health parameter in the game which can then be called a health evolution (he).

$$f_2 = \left(\frac{(150*x_1+0.7*x_3+0.5*x_4)}{3}\right) * \frac{x_2}{w} * \left(\frac{\left(\frac{x_1}{x_3}-x_4\right)}{3}\right) * n * he + \left(\frac{hp*NM*dp}{2}\right) \quad (9)$$

Where:

gh = gen health
he = health evolution

$$w = x_1 + x_2 + x_3 + x_4 \quad (10)$$

Where:

w = weight
x_1 = genScore
x_2 = genHealth
x_3 = genNpcMiss
x_4 = genDamage

4. EXPERIMENT

In this paper, we examine the dynamic evolutionary of the NPC speed and health using NSGA II which is doing by looking at the objective function of the player such as player score, player health, NPCmissed, and player damage in order to optimize the dynamic evolution of the NPC on the objective function as speed(f_1) and health(f_2) to find appropriate changes of the fitness value according to find the flow zone in the game.

4.1 Experimental on Game Testing for Beginner Player

Game testing on the beginner player is used for getting the data player in one game while facing the dynamic evolution of the NPC speed(f_1) and health(f_2)at each level. In order to obtain the data to determine flow zone for the beginner player in one game.

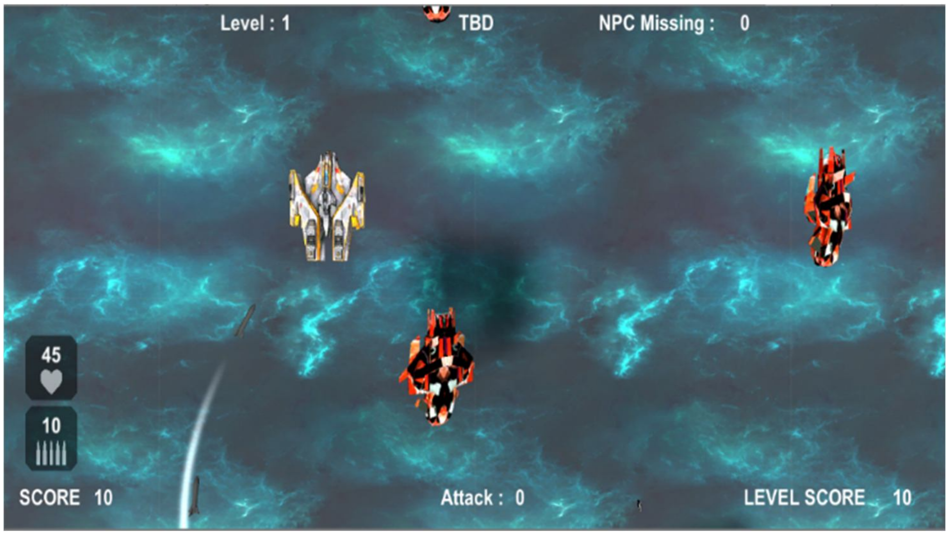


Figure 8. Game Testing for Beginner Player

During the player plays the game, the data obtained to determine the changing parameters of the player during the game plays in each level that has been passed by the player. The parameters of the data will be processed as NSGA II. To find the best solution in finding a flow zone for the beginner player. The data parameter of the beginner player is shown in Table 8.

Table 8. Data of Beginner Player in One Game.

Game	Level	Level Game	Player health	Player Score	NPC Missed	Player Damage
1	1	default	50	100	20	93
2	2	R_2	65	200	10	125
3	3	R_3	70	300	13	80
4	4	R_5	0	380	5	310
5	1	R_1	100	100	0	230
6	2	R_7	0	150	5	313
7	1	R_1	-	-	-	-

4.2 Experimental on Game Testing for Expert Player

Game testing on the expert player is used for getting the data player in one game while facing the dynamic evolution of the NPC speed(f_1) and health(f_2)at each level. In order to obtain the data to determine flow zone for the expert player in one game.

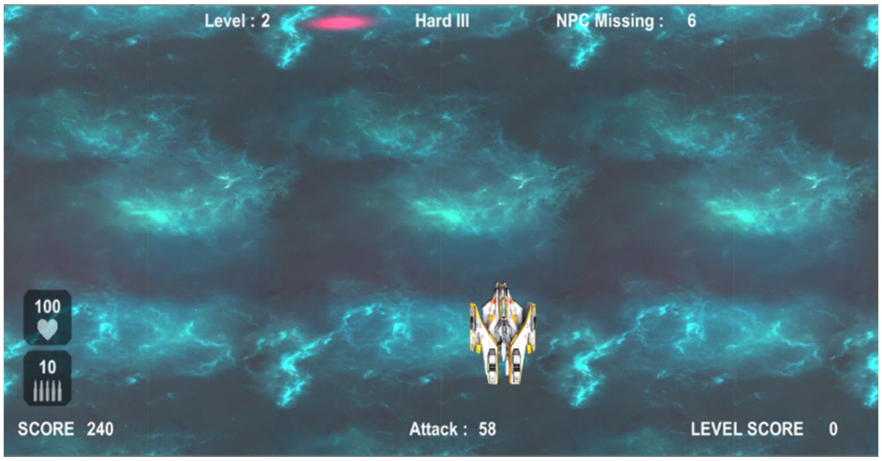


Figure 9. Game Testing for Expert Player

During the player plays the game, the data obtained to determine the changing parameters of the player during the game plays in each level that has been passed by the player. The parameters of the data will be processed as NSGA II. To find the best solution in finding a flow zone for the expert player. The data parameter of the expert player is shown in Table 9.

Table 9. Data of Expert Player in One Game

Game	Level	Level Game	Player health	Player Score	NPC Missed	Player Damage
1	1	default	75	100	2	50
2	2	R_5	30	200	5	160
3	3	R_3	85	300	11	230
4	4	R_7	90	400	7	490
5	5	R_9	0	430	15	320
6	1	R_1	100	100	4	30
7	2	R_8	-	-	-	-

4.3 Testing NSGA II Multi-Objective With Two Leveling Behaviour

In the experiment, the results of the experiment NSGA - II for multi-objective on the dynamic evolution of the NPC on objective function to improve the function of speed(f_1) and health(f_2) of the NPC.

Genetic algorithm parameters used in this test was as follows:

1) The number of decision variables (n) = 10.
2) The population size = 50.
3) The generation size = 100.
4) The probability of crossover (p_c) = 1. Index of crossover distribution (n_c) = 20.
5) The probability of mutation (p_m) = 1/n. Index of mutation distribution (n_m) = 20.

The result of the experiment NSGA-II In the dynamic evolution of the NPC have been able to find Pareto optimal solution which is good enough for simulating the dynamic evolution of the NPC. Which is the optimal solution obtained in compliance with the Pareto optimal in trials with 50 populations and 100 generation. Pareto sets have shown the rate of the changing on Pareto-optimal front which is a set of solutions in the testing objective function to get the fitness value corresponding to the flow zone. By using adaptive mutation in the dynamic evolution of the NPC, it can be repaired average fitness value significantly compared to previous generations to get some optimal solutions (multi-objective solution). In Figure 8 shows a graph that showing the population distribution or the changing of Pareto set on the fourth generation to achieve Pareto optimum in speed(f_1) and health(f_2).

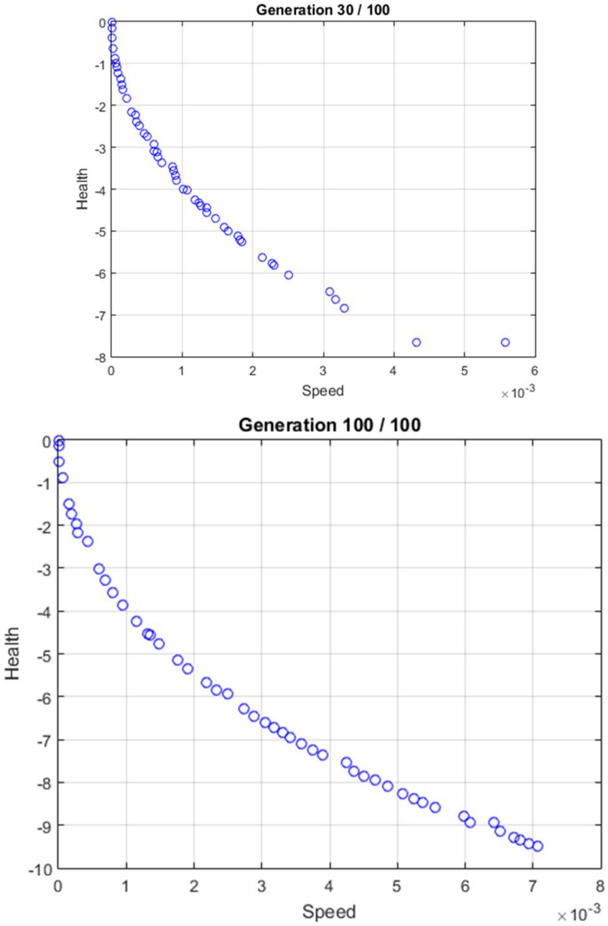


Figure 10. Experiment within 50 populations dan 100 Generation

In figure 8 represent the result of increasing in the spread of objective function of the NPC at the speed (f_1) which is considerably higher than the objective function of health (f_2). The value of function player score have a inversely proportional with the value of function player health, while the value of function NPCs missed have directly proportional with the value of player damage.

Table 10. Fitness Value Within 50 Populations and 10 Generation

Data Ke	Gen Score	Gen Health	Gen NPC missed	Gen Damage	Health NPC	Speed NPC
91	0.486931	0.830226	1	1	0.507952	-8.26217
92	0.474437	0.680488	0.999773	1	0.359138	-7.10856
93	0.490781	0.418875	1	1	0.147988	-4.75336
94	0.490102	0.984112	1	1	0.681015	-9.35261
95	0.456657	0.397279	1	1	0.135779	-4.56879
96	0.461111	0.365211	1	1	0.115949	-4.24659
97	0.469212	0.009602	1	1	1.225949	-0.12756
98	0.491706	0.874949	1	1	0.555744	-8.58138
99	0.432464	0.903871	1	1	0.598409	-8.80408
100	0.476307	0.393636	1	1	0.132567	-4.52233

The optimization process that happens after conducting the test is used for getting the value of the objective function according to the player's ability so it can find the optimization objective function of the NPC. In order to optimize the dynamic evolution of the NPC to be able to face an opponent.

5. CONCLUSION

Based on the condition of the owned players, the conditions that exist on the player, used as a parameter and will be compared with each other, then used as input to the NSGA-II to optimize the selection of NPC dynamic evolution by looking at some of these parameters.

In this research, the four parameters are used, there are player score, player health, NPCmissed and player damage of the player who used to kill the NPCs.

Optimal behavior that was developed is dynamic evolution of the NPC that can evolve the fitness function of the NPC such as speed and health.

From the test data that has been done in this research, and the test results that evidenced by the graph that showing the spread of the Pareto chart which indicates a set of pareto optimal stability of the game where the font of pareto optimal has achieved the best stability in order to obtain significant data and find a solution to multi-objective dynamic evolution of the NPC.

The overall of optimal behavior that done having the objective of maximizing dynamic evolution of the NPC to face the player. So the NPC has the adaptive ability to deal with the changing behavior of the player.

REFERENCES

[1] M. J. Osborne. An Introduction to Game Theory, MIT Press, 1994

[2] Adhityo Dewandoro, Genre (Tipe) permainan pada Games. Universitas Gunadarma. 2016

[3] Cindy Taurusta (2014), Optimasi Untuk Perilaku Pahlawan Berbasis NSGA – II. International Conference on Information

[4] hen, J., (2007), “Flow in Games (and Everything Else)”, Communications of the ACM Viewpoint Paper, hal 31 – 34

[5] Ariyadi (2014), Creep Offenses Evolution in Tower Defense Games using NSGA-II, International Conference on Information

[6] Abraham, A., Jain, L., & Goldberg, R., (2005) Evolutionary multiobjective optimization: Theoretical Advances and Applications. London: Springer

[7] Gen M, and ChengR, Genetic Algorithms and Engineering Design, New York: John Wiley & Sons, Inc. 1997.

[8] Gen M, and ChengR, Genetic Algorithms and Engineering Optimization, New York: John Wiley & Sons, Inc. 2000.

[9] Lozano M, and HerreraF, Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy,Soft Computing. 7: 545–562. 2003.

[10] Lozano, M., F. Herrera, N. Krasnogor, and D. Molina, Real-Coded Memetic Algorithms with Crossover Hill-Climbing. Evolutionary Computation **12**(3): 273-302, 2004.

[11] Lozano M, HerreraF, KrasnogorN, and MolinaD, Real-Coded Memetic Algorithms With Crossover Hill-Climbing, Evolutionary Computation 12(3): 273-302, 2004.

[12] Jaszkiewicz A, Genetic Local Search for Multi-Objective Combinatorial Optimization, European Journal of Operational Research, 137(1) : 50-71, 2002.

[13] Mahmudy WF, Optimasi Fungsi Tanpa Kendala Menggunakan Algoritma Genetika dengan Kromosom Biner dan Perbaikan Kromosom Hill-Climbing, Kursor, 4 (1) : 23-29, 2008.

[14] Deb K, Agrawal S, Pratap A, Meyarivan T. (2002), "A fast and elitist multi-objective genetic algorithm : NSGA-II", IEEE Transactions on Evolutionary Computation, Vol. 6 No. 2, hal. 182–197.

[15] Liu, C., Agrawal, P., Sarkar, N., & Chen, S., (2009), "Dynamic Difficulty Adjustment in Computer Games through Real-time Anxiety-based Affective Feedback". International Journal of Human–Computer Interaction, Vol. 25, hal. 506 - 529.