Modified Flocking Algorithm for Optimizing Non Player Character Movements

Astria Sastra Dewi[1], Eka Wahyu Hidayat[2], and Heni Sulastri[3]

[1]Departement of Informatics, Faculty of Engineering, Siliwangi University
astriasastra@gmail.com
[2]Doctor of Engineering Study Program, Udayana University, Bali
ekawahyu@unsil.ac.id
[3]Departement of Informatics, Faculty of Engineering, Siliwangi University
henisulastri@unsil.ac.id

Abstract Non player character is an important part of a game that can make gameplay more interesting and challenging. That reason also makes it important to study, especially for non player characters that are clustered in large numbers where programming of such non player characters is relatively difficult to do. Quite a lot of games with non player characters in the form of birds that are clustered but on average the game does not apply good movement so that the motion of birds is relatively slow, monotonous and unrealistic. In this research, we will try to modify the Flocking Algorithm, as one of the popular algorithms for non player character groups, so that birds can fly scattered in terrain while maintaining flocking. In its implementation, the new algorithm was successfully designed and inserted in flocking, which was then applied to a 3D game. The test results are quite satisfactory with only 6.6% collisions on 100 agents with a maximum frame rate of 10 fps. Non player characters succeeded well toward 3 targets in less than 2 minutes and the results of beta testing were an average of 81% of respondents "VERY AGREE" The implementation of the Flocking Algorithm and its modification worked well and resulted in a realistic movement of non player characters.

***Index Terms*— Artificial Inteligence , Flocking , Game, Non Player Character.**

I. INTRODUCTION

Various types of games have grown and developed in Indonesia. Some things that become the attraction of a game and can make players enjoy playing the game is the quality of the game in terms of graphics and gameplay. Gameplay is an important part of a game because it is the core of the game itself. The part that cannot be separated and very closely related to gameplay is the movement of non player characters in the game, especially for the first person shooter genre game and real time strategy.

Non Player Character (NPC) is a character in a game that moves and behaves without control from the player and only moves according to the program given to him. NPC is a very important component for modern computer games that will determine whether the game is interesting or not [1]. NPC programming can be done by embedding simple movements and selection programs if there is only one NPC, but if there are NPCs grouped in one game in the same scene and time, then programming on the NPC will not be efficient and will be difficult to do manually so it requires artificial intelligence.

One of the popular artificial intelligence to be applied to grouped NPCs is the Flocking Algorithm. This algorithm is also called the Boids Algorithm. Flocking have been tested on a group of bird objects that have successfully moved in groups in a realistic behavior [2]. However, flocking have a weakness, which is only providing regular movements without having a purpose. This algorithm cannot find which path to pass so it has a tendency to get stuck in complex fields [3].

In this research, what will be done is to make the NPC group move towards the specified target by first recording all targets, but still maintaining the group process. Therefore an experiment will be carried out to modify the flocking and be implemented in 3D games in the hope that the NPC will be able to go to certain places and remain in the flocking condition.

II. LITERATURE REVIEW

A. Non Player Character

Non player character or often called NPC, is a type of autonomous agent intended for the use of computer animation and interactive media such as games and virtual

reality [4]. In games, NPCs cannot be controlled by players [2] [5]. The movement of the NPC came from the program that was applied to it. The program can be a simple program or a complex program (Artificial Intelligence). The game will look more realistic if artificial intelligence is applied to the character, especially on the NPC [6]. Non Player Character can be applied as an enemy character or a friend's character in a game [7].

B. *Flocking Algorithm*

The Flocking Algorithm is often called the Shooting Bird or the Reynolds Algorithm, because the first to introduce it was Craig Reynold in 1986. This algorithm was created based on the movement of flocks of birds, herd of land, or fish herds [2]. This algorithm functions to give movement to a clustered object so that its movement resembles a crowd of real animals [2], and reacts only to flockmates in their own environment [8]. Flocking means a crowd, while one object that is in a crowd is called a boid. Each boid in the crowd must know the position and speed of the boids in the vicinity (neighbor) [9]. Every time the crowd moves, the position of a boid will change constantly so that each boids must update neighbor information during the game [9]. Reynold mentions three rules that must exist to create a group movement [2], namely:

1) *Collision Avoidance*: avoid collisions with the closest flockmates / neighbors
2) *Velocity Matching*: try to match it with the closest flockmates
3) *Flock Centering*: an effort to stay close to the closest flockmates

These three rules are better known as separation / avoidance, allignment, and cohesion.

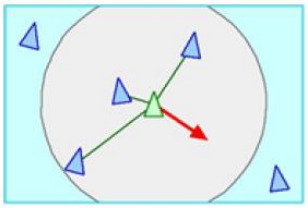

Fig. 1. Separation

Figure 1 explains the separation rule which means the ability to control behavior in maintaining or maintaining the distance between adjacent boids so as not to collide or density [10]. The separation rules can be calculated using the following equation:

$$d(Px,Pb) \leq d1 \cap d(Px,Pb) \geq d2 \rightarrow Vsr = \sum_x^n \frac{Vx+Vb}{d(Px,Pb)} \quad (1)$$

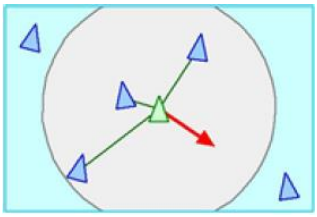

Fig. 2. Allignment

Figure 2 explains the alignment rules, namely the ability to control behavior to align with the surrounding agents, aiming to go in the same direction and match the speed of each agent in the herd [10]. The alignment rules can be calculated using the following equation:

$$d(Px,Pb) \leq d1 \cap d(Px,Pb) \geq d2 \rightarrow Var = \frac{1}{n}\sum_x^n \frac{Vx+Vb}{d(Px,Pb)} \quad (2)$$

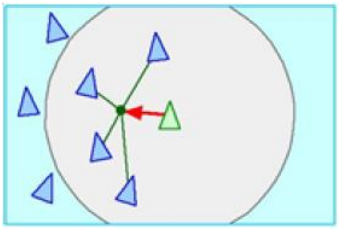

Fig. 3. Cohesion

Figure 3 explains the cohesion rule that is the ability to control behavior to move toward the average position of the closest herd [10]. Cohesion rules can be calculated using the following equation:

$$d(Px,Pb) \leq d1 \cap d(Px,Pb) \geq d2 \rightarrow Pavg = \sum_x^n \frac{Px}{n} \quad (3)$$

III. METHODOLOGY

A. *Data Collection*

The data collection process will be carried out in two ways, namely a literature study that collects information about several matters relating to research and observation to find out similar games related to research.

B. *Development*

This stage will be made game design and construction which will be the object of research. Game development will refer to Luther's version of multimedia development methodology [11].

C. *Testing Algorithm Implementation*

This test will be seen from 4 criteria, namely the number of collisions, frame rates, validation of objects that reach the target and feedback from end users.

IV. RESULT

A. *Data Collection*

1. Literature Study

Craig Reynold, in 1986, became the first person to introduce his algorithm to make clustered objects mimic the movements of bird flocks. Reynold argues that traditional computer animation that regulates the location and movement of objects is considered inefficient to be applied to a large number of clustered objects [2]. Meanwhile flocking research applied to games was conducted by [12] who applied flocking to NPC movements in 2D games with endless run genre. According to him, the existence of varied obstacles in the game is very important to make the game not

monotonous. In a study conducted by [10], flocking was applied so that several birds did not collide while flying, when one bird was successfully shot by a player who was stationary in one place. The test is done by observing the coordinates of 7 birds that are applied to the Flocking Algorithm. The result was that not a single bird collided.

2. Observation

Observations were made by examining several FPS games that have bird NPCs. There are 5 games studied, namely Birds Hunting, Birds Shooting, Pheasant Shooter, Dessert Bird Sniper and Duck Hunting 2019. The five games have almost the same concept. Players in the game cannot move and can only shoot and aim shots. NPC birds in the game are relatively few in number and fly at relatively slow speeds. Of the five games, only the Pheasant Shooter whose NPC bird has the ability to adjust the distance and direction of flight. But it still has a weakness, namely after a few minutes the NPC will pile up. Below is a picture of the game Pheasant Shooter whose NPCs are piled up:

Fig. 4. NPC Game from Pheasant Shooter is Piling Up

B. *Development*

1. Concept

The concept of the game that will be made in this study is in table 1 below:

TABLE I
THE SHOOTING BIRD GAME CONCEPT

Title	Shooting Bird
Audience	The particular person chosen to be a tester
Duration	Limited until the system declares win or game over
Audio	Backsound (* .mp3)
Image	Background menus, buttons, icons (* .jpg)
Text	Text on buttons, instructions, captions in the game and test captions
Animation	3D character animation
Interactivity	Button and analog controller
Theme	Shoot the bird the number ordered by the system
Character	Consisting of 1 character and many NPCs
Application Description	This game was created to test the Flocking Algorithm and its modifications to be applied to NPC collections. The rules of the game are that the player must shoot the number of birds ordered by the system. Gameplay is made simple because it focuses on NPC movements and interactions between players and NPCs.

2. Design

a. Gameplay

In short the Shooting Bird game will have the following gameplay:

1) The player determines the number of bird NPCs
2) After entering the main game, the player must control the character and shoot the bird a number of commands in the game
3) If the shot is missed by the NPC then the bird will attack the player. Bird attacks can be avoided by hiding under trees
4) There is no time limit, the game ends when the number of birds shot is as per the command or the player's life has run out

b. Design of Flocking Algorithm and its Modifications

Based on the basic rules of the Flocking Algorithm namely cohesion, alignment and avoidance, the overall Flocking Algorithm applied in the Shooting Bird game is illustrated in the flowchart in Figure 5:

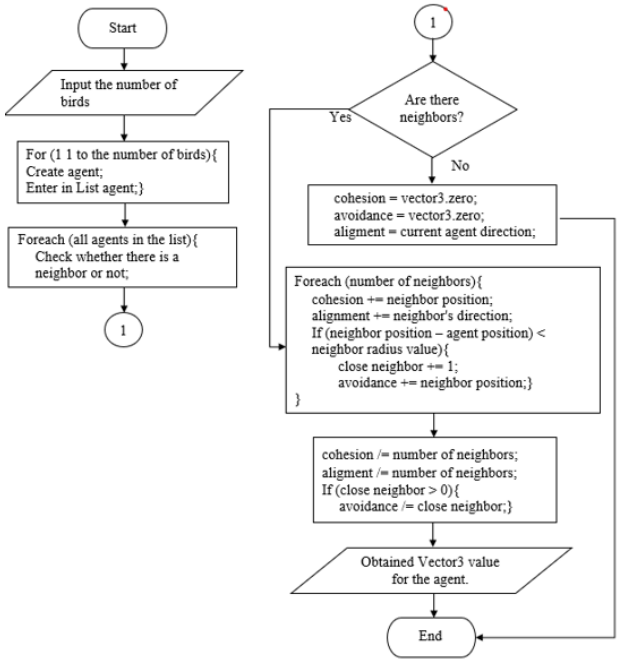

Fig. 5. Flocking Algorithm Flowchart

Subsequently modifications were made to the flocking so that the boids could be spread to 3 specific places while maintaining the flocking condition of the herd. Modifications are made by creating a new algorithm that is inserted into the flocking, and serves to create three leaders from all the flock to find a place and become a leader towards that place. In terms of finding a place, the algorithm used is similar to an existing search algorithm, the djikstra algorithm. The similarity is that both will check all the paths connected to the starting place to choose the shortest path. But after further investigation, the algorithm is purely to find a path to get to one goal, as explained in the study [13] which states that the Djikstra algorithm aims to find the shortest path based on the smallest weight from one point to another. This is different from that applied in this study, where the algorithm will choose the shortest distance from 3 places, then go to that place. If it has been visited, the place will not be checked again and the search will be carried out in the remaining two places, and so on until all places have been successfully visited. When finished, the place visited will be reset and the search will be repeated from the beginning by the new leader. This explanation proves the difference between the Djikstra

algorithm and the flocking modification algorithm in this study. Therefore, to facilitate discussion, this new algorithm will be named the Aster Algorithm. Next is the Aster Algorithm flowchart in figure 6:

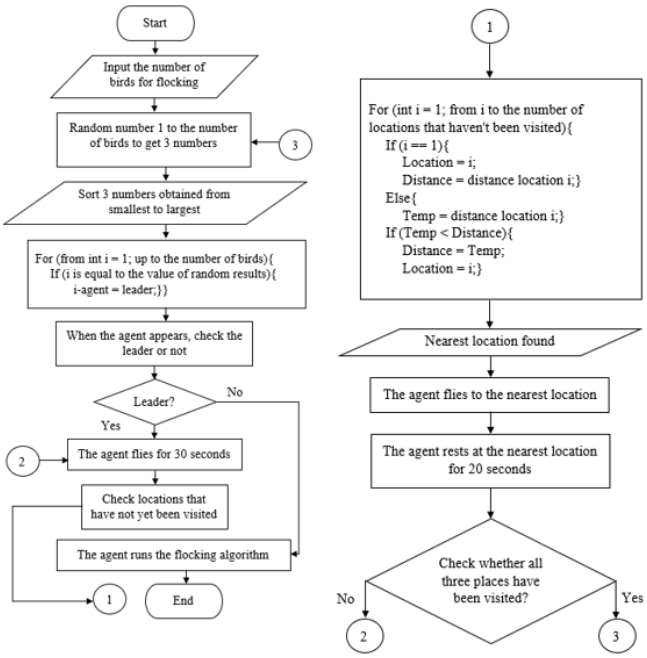

Fig. 6. Aster Algorithm Flowchart

3. Collecting Content Material

At this stage the process of finding or making materials needed in the game is carried out, such as pictures, music and animation. image editing process using Corel Draw X7.

4. Assembly

At this stage a combination of all material and concepts that have been made in the previous stage is made so that it becomes a whole game. Below are some Shooting Bird game interface in figure 8-10:

Fig. 7. Game Initial Display

Fig. 8. Game Preparation Page

Fig. 9. Main Game Display

Fig. 10. Two Flying Birds Hovering Above the Target Tree

The game engine used is Unity and the process of making program code using the C3 programming language in Visual Studio Code 2019.

5. Testing

Tests that will be carried out at this stage are testing from the developer side or alpha test which is made to see the possibility of errors in the system development and algorithm implementation. The method used is blackbox testing which functions to see whether the game is running according to design or not. From all test scenarios, the application response shows that it has met the expected results.

6. Distribution

Shooting Bird game that has been completed and has passed the alpha test will be distributed by uploading it to Google Drive, the goal is that selected tester candidates whose position is far from the researcher can download the Shooting Bird game by themselves for testing.

C. Testing the Algorithm Implementation

This test was carried out on smartphone devices using Android version 7.0 NRD90M, and hardware with a 2.0 GHz Snapdragon 625 octa-core 2.0 processor, 3.00 GB RAM, and GPU Adreno 506.

The testing procedure is:

a. Run the game that has been installed on the device
b. Inputed 100 birds for NPC
c. The game is left running for more than 3 minutes
d. The results that appear in the test info are recorded
e. The game is stopped, return to the main menu and input 200 birds for the NPC
f. Points c, d and e are repeated by inputting 300 NPCs
g. After it is finished, the game is stopped, RAM is cleared and the 2nd experiment starts in the order of points a through f
h. After the 2nd experiment is finished, proceed with the 3rd trial

The test results are in table 2:

TABLE 2
TEST RESULTS OF ALGORITHM IMPLEMENTATION

Trial to-	Number of Agents	Minute to-	Average frame rate	Number of Collisions	Time to All Targets
1	100	1	9	3	1 minute 51 seconds
		2	10	2	
		3	10	0	
	200	1	7	19	1 minute 43 seconds
		2	9	8	
		3	9	4	
	300	1	3	2311	1 minute 55 seconds
		2	6	171	
		3	7	39	
2	100	1	9	4	40 seconds
		2	10	3	
		3	10	1	
	200	1	5	33	1 minute 44 seconds
		2	8	18	
		3	8	5	
	300	1	4	3509	1 minute 58 seconds
		2	6	60	
		3	6	49	
3	100	1	9	4	46 seconds
		2	10	2	
		3	10	1	
	200	1	7	14	45 seconds
		2	9	9	
		3	9	3	
	300	1	3	3264	1 minute 52 seconds
		2	6	72	
		3	6	51	

The results of collision testing showed that from 3 times the average collision test on 100 agents was 6.6% or only happened 6 to 7 collisions for 3 minutes. This result can be said to be very good when compared to the flocking implementation research conducted by [12] which only applied and tested the Flocking Algorithm at a maximum of 8 NPC 2D birds and [10] which only on 7 3D birds. For frame rate testing, the results obtained are less than the maximum because the highest frame rate is only up to 10 fps, while the ideal FPS game is at 30 fps. This also causes a lot of collisions to occur in tests with a total of 200 and 300 NPCs. For testing NPCs towards the target, the NPC managed to get to the 3 targets in less than 2 minutes, which indicates that the algorithm implemented went smoothly.

D. Beta Testing

Beta testing is done with the aim to validate the usability, functionality, compatibility and reliability testing of the games made [14]. This test was conducted with a purposive sampling technique with a total of 12 questions packaged in a Likert scale questionnaire [15]. The results show that from an algorithm side an average of 81% of respondents "VERY AGREE" The implementation of the Flocking Algorithm and its modification worked well and resulted in a realistic and average NPC movement overall, 79% of respondents "AGREE" The Shooting Bird Game went well.

V. CONCLUSION

After identifying the problem, the design and implementation of the system obtained some conclusions as follows:

1. The design and construction of 3D Shooting Bird Games for Android has been successfully carried out according to the concept with relatively many bird NPCs and players who can explore the game arena or terrain, both of which have never been included in previous studies.
2. The application of Flocking Algorithm was successful and a new algorithm was created to add features to the flocking, the Aster Algorithm which functions to make the NPC able to reach the target that has been determined so that the NPC can be spread.
3. The test results are quite satisfactory with only 6.6% collisions on 100 agents with a maximum frame rate of 10 fps. The NPC succeeded well towards 3 targets in less than 2 minutes and the results of beta testing were an average of 81% of respondents "VERY AGREE" The implementation of the Flocking Algorithm and its modification worked well and resulted in a realistic NPC movement.

REFERENCES

[1] Y. A. Benufinit, M. Hariadi and S. M. S. N, "Manuver Kelompok NPC Berbasis Boids," in *Seminar Nasional ke – 9: Rekayasa Teknologi Industri dan Informasi*, Yogyakarta, 2014.

[2] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, pp. 25-34, 1987.

[3] J. Hagelback, "Hybrid Pathfinding in StarCraft," *IEEE Transaction On Computational Intelligence and AI in Games*, pp. 319-324, 2016.

[4] M. B. Nendya, S. G. Gunanto and R. G. Santosa, "Pemetaan Perilaku Non-Playable Character Pada Permainan Berbasis Role Playing Game Menggunakan," *Journal of Animation and Games Studies*, vol. 1, no. 2, pp. 185-202, 2015.

[5] Atthariq and D. A. Putra, "PENENTUAN PERGERAKAN NON-PLAYER CHARACTER MENGGUNAKAN ALGORITMA A* PADA GAME ACTION- ROLE-PLAYING GAME," *Jurnal Infomedia*, vol. 2, no. 2, pp. 35-40, 2017.

[6] S. Karim, "Perubahan perilaku Non-Player Character (NPC) pada Game Arabic Hunter menggunakan Jaringan Syaraf Tiruan Perceptron," *Jurnal Ilmiah Teknologi Sistem Informasi*, pp. 34-41, 2017.

[7] A. Sugito, M. Hariadi and A. Z. Fanani, "PERGERAKKAN ADAPTIF NONPLAYER CHARACTER TEMAN BERBASIS EVENT MENGGUNAKAN BEHAVIOR TREE DAN RULE BASED SYSTEM," *Jurnal Teknologi Informasi*, vol. 13, no. 2, pp. 84-91, 2017.

[8] A. Sikenali and Atthariq, "ANIMASI 3D PERGERAKAN BURUNG TERBANG MENGGUNAKAN MODEL BOIDS," *Jurnal Infomedia*, vol. 1, no. 2, pp. 1-5, 2016.

[9] L. M. Ginting, B. Siahaan, B. Situmorang and R. Manik, "KAJIAN ALGORITMA CRAIG RAYNOLD PADA KERUMUNAN (FLOCKING)," *Jurnal ISD*, pp. 22-32, 2018.

[10] D. R. Harefa and M. B. Ismiati, "PENERAPAN ALGORITMA BOIDS PADA NPC (NON PLAYER CHARACTER) DALAM GAME," in *Seminar Nasional Teknologi Informasi dan Komunikasi 2019 (SENTIKA 2019)*, Yogyakarta, 2019.

[11] M. F. Azim, E. W. Hidayat and A. N. Rachman, "Android Battle Game Based on Augmented Reality with 2D Object Marker," *JOIN (Jurnal Online Informatika)*, vol. 3, no. 2, pp. 116-122, 2018.

[12] Yosua, E. M. A. Jonemaro and M. A. Akbar, "Penerapan Flocking Behavior Untuk Pergerakan Berkelompok Non Player Character pada 2D Endless Runner Game," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 4178-4184, 2019.

[13] D. Ardana and R. Saputra, "Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus," in *Seminar Nasional Ilmu Komputer*, Semarang, 2016.

[14] E. W. Hidayat, A. P. Aldya and P. T. A. Miranti, "Game Adu Muncang Berbasis Android," *Innovation in Research of Informatics (INNOVATICS)*, vol. 1, no. 1, pp. 13-20, 2019.

[15] Sugiyono, Metode Penelitian Kualitatif, Kuantitatif dan R&D, Bandung: Alphabeta, 2012.