

Generation of space shooter level using genetic approach

Ahmad Hamdani^{a,1,*}, Wahyu Andhyka Kusuma^{b,c,2}

^a Universitas Muhammadiyah Malang, Jl. Tlogomas No.246, Malang dan 65144, Indonesia

^b Universitas Muhammadiyah Malang, Jl. Tlogomas No.246, Malang dan 65144, Indonesia

¹ hamdaniryuiken@gmail.com *

* corresponding author

ABSTRACT

In this article, we used genetic algorithm and geometric-based approach to generating level for 2D space shooter game. We used the defined fitness value from game designer to limit the fitness value of the genetic algorithm process. And the geometric-based is used to generate base level from the best chromosome in genetic algorithm. The geometric generator will take a random object for each corresponding game element from the chromosome. This approach minimize the time to generating the object, we directly used object geometry for the data in chromosome so it can minimize process and memory cost. This approached minimizing the content that must be created manually. From the result, to generation level with controlled difficulty, we must change the chromosome length too, so the fitness value can fit the target fitness and not show any linear difficulty. Our result showed this method capable of generating different level and controlled difficulty.

Keywords:

Artificial Intelligence
Games
PCG
Genetic Algorithm
Evolutionary Algorithm

I. Introduction

Game world is an important aspect in Game. Game world is artificial world, an imaginary place where the game event occur [1]. Mostly, game world is designed by game or level designer. This is not a big problem for small scale games. However, if the game scale is growing bigger it will take more time and resources. Game development required quite a lot cost and time [2]. While, it requires a few time to play it. This is one of the trap when developing new game, the lack of ability to interest the player to play the game again and make the game can not last longer [3]. Procedural content generation aims to generate content automatically and reduce the cost of design and game development [4][5][6]. Procedural content generation (PCG) is game content that created by algorithm with limited or indirect input from user. In other words, PCG refers to software that able to create content alone or together with one or more human players or designers [7]. By using PCG, the game content will be richer. Such as, every time player play new game, the game world will be different. This will provide the player with challenge and appeal. The presented method is aimed to implementing PCG using genetic algorithm and grammar-based approach on top down shooter game. Using the chromosome data from genetic algorithm and creating level base with grammar approach, resulting in a level with set able difficulty and different level.

II. Method

The approach for generating level described in Fig. 1. It start with a small random chromosome for initialize genetic algorithm. The genetic algorithm will process the chromosome and return the best chromosome. The level generator will take the best chromosome to perform base level generator. to form base level, chromosome or game element generated and evaluated using genetic algorithm. Base level created using the data from the Genetic Algorithm. After generating the base level, the level will decotate with some collectable object. In this article, we choose health. In this game, health is a perfect example of collectable object. The parameter of genetic algorithm and level generator is set by human designer. Parameters include population size, chromosome size, elitism, game element, range each element, rest area, target fitness. Rest platform is an empty space before game element appear.



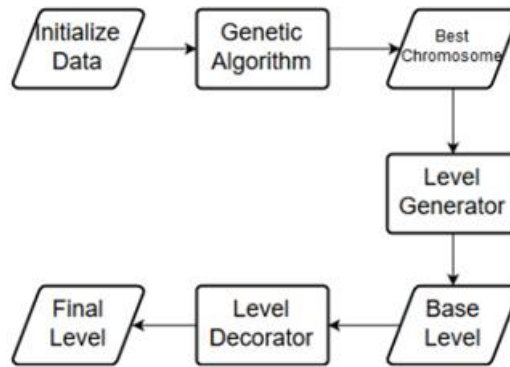


Fig. 1. System Overview

III. Result and Discussion

A. Generation of Chromosome

The chromosome is just an array data structure that holds the data of geometry for generating the game world. It shows which geometry will be used in the game world. To form the chromosome, first we need mapping from language to numbers. Table 1 shows this mapping for each game element.

Table 1. characteristic of the datasets

Element	Number
0	Meteor
1	Meteor Canon
2	Enemy Ship

B. Genetic Algorithm

Genetic algorithm has been chosen to create the chromosome. Using this method, there is no need to filter the generated chromosome, because this algorithm will produce polished chromosome, not just some random ones (Fig. 2).

0	0	1	0	2	1
---	---	---	---	---	---

Figure 2. Example of chromosome.

Choosing the right fitness function is crucial in genetic algorithm to work properly. There is need function that tell us which one is good and not. So, the target fitness can be defined by curve or direct value from designer. The target fitness value will be the threshold for the fitness calculation. If the fitness value if fullfil the target fitness, the algorithm will stop. To calculate fitness value, we give hardness value between transition of elements (Table 2). This value can be change by game designer to form better level in the game

Table 2. Hardness Value Between Element

Previous Element	Next Element	Hardness Value
	Meteor	
Meteor	Cannon	1
Meteor	Enemy Ship	1
Meteor Cannon	Meteor	1
Meteor Cannon	Enemy Ship	2
Enemy Ship	Meteor	1
	Meteor	
Enemy Ship	Cannon	2

According to the Table 2 hardness value with the highest value is when player need to perform more than one action, in example avoiding bullet or enemy and shooting the enemy ship. We calculate the sum of hardness value for the fitness value.

After creating the chromosome, to generate playable level based on the chromosome. The chromosome is passed into level generator. For given element, many objects can be implemented, but here we just use one object for each game element. Random game object will be chosen to generate a new level and the position of the object will be adjusted automatically based on element index in chromosome. The collectible item will be placed in random place based in range between first game element and last game element. Shows in Table 3.

Table 3. Element & the Corresponding Geometries.

Element	Corresponding Geometries
Meteor Cannon	Set of Cannon in planet of meteor
Meteor	Set of meteor
Enemy Ship	Set of enemy ship

Unity3D and C# is used to implement the proposed method. There only 3 element in this implementation, we can add more element to the game by adding more element and corresponding geometries. We want to start game from the easy one so we used the chromosome with the lowest fitness value. Fig.3 shows an example of level that generated by the proposed method.

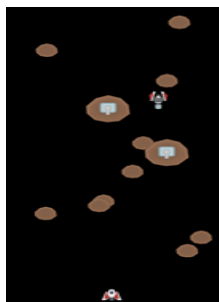


Figure 3. An example of generated level with target fitness 4.

We limit the population to 10 and stop the algorithm if there is no change in fitness value for 10 generation. From the result, there is still high error from target fitness an actual fitness, shows in Table 4.

Table 4. Evaluation Result

No	Target Fitness	Gene	Actual Fitness
1	1	0112101001	6
2	2	1010120210	6
3	3	0210011010	5
4	4	0010020101	4
5	5	1201100002	5
6	6	0101101210	6
7	7	2021120110	5
8	8	1212100010	6
9	9	1012121001	7
10	10	1010010112	6

However, the generated level and difficulty still can be controlled by changing the parameter value. For the next result we increase the population size by to for each target fitness and it shows in Table 5. The comparison between default parameter and modified parameter shown in Fig. 4, show if the change in chromosome length is affecting the actual fitness.

Table 5. Evaluation Result After Modification.

No	Target Fitness	Gene	Actual Fitness	Gene Length
1	1	2121	2	4
2	2	211221	4	6
3	3	21011212	8	8
4	4	0012020122	4	10
5	5	121210121212	11	12
6	6	12011021011012	9	14
7	7	1212121201012112	13	16
8	8	021212122110021211	12	18
9	9	02120110122221221102	9	20
10	10	2112120112212112121012	17	22

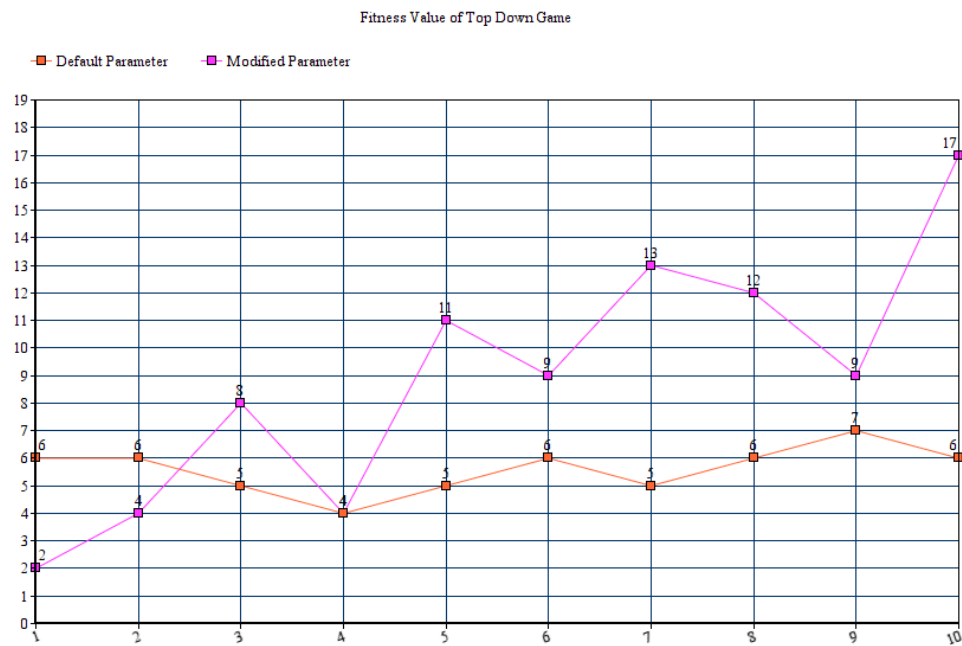


Figure 4. Comparison between default parameter and modified parameter

From the comparison, the modified parameter perform better result in generatin new level, the value is not decrease excessive. So the level can give a rest level to the player after playing some level if the hardness value is less than the previous value.

IV. Conclusion

In this article, we proposed a method to produced a generated level with difficulty value by human game designer. A prototype has been developed. Genetic algorithm has been used to created the chromosome of element with fitness function which is the sum of hardness value. Level Generator generate the corresponding object randomly based on the grammar. In the future work, we decide to normalize the fitness value, add new element, and change the placement of the object generate by level generator.

Acknowledgment

The author would like to say a lot of praise to Allah SWT who has provided ease for authors in completing this journal for the smooth task of seminar courses and Scientific Writing. Also thanks to friends who have supported the author in completing this research.

References

- [1] E. Adams, *Fundamentals of Game Design 2nd Edition*, 2nd ed. New Riders, 2010.
- [2] B. Li and M. Riedl, "An Offline Planning Approach to Game Plotline Adaptation," *Assoc. Adv. Artif. Intell.*, pp. 45– 50, 2010.
- [3] J. Ulisses, R. Gonçalves, and A. Coelho, "Procedural Generation of Maps and Narrative Inclusion for Video Games," *Oporto*, January, no. June, 2015.
- [4] A. Amato, "Procedural Content Generation in the Game Industry," *Springer Int. Publ. AG 2017*, pp. 15– 25, 2017.

- [5] D. M. De Carli, F. Bevilacqua, C. T. Pozzer, and M. C. D'Ornellas, "A survey of procedural content generation techniques suitable to game development," *Brazilian Symp. Games Digit. Entertain. StablBGAMES*, pp. 26–35, 2011.
- [6] J. Roberts and K. Chen, "Learning-based procedural content generation," *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 1, pp. 88–101, 2015.
- [7] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games*. Springer International Publishing, 2016.