Procedural Generation of Angry Birds Levels That Adapt to the Player's Skills Using Genetic Algorithm

Misaki Kaidan*, Chun Yin Chu†, Tomohiro Harada*, and Ruck Thawonmas*
Intelligent Computer Entertainment Laboratory
*College of Information Science and Engineering
†Graduate School of Information Science and Engineering
Ritsumeikan University, Shiga, Japan
ruck@is.ritsumei.ac.jp

***Abstract*— This paper proposes a procedural generation method that automatically creates game levels for Angry Birds, a famous mobile game, using genetic algorithm. By adjusting the parameters of the genetic algorithm according to the player's gameplay results, our proposed method can generate game levels that adapt to the player's skills. Our experiment proves that the proposed method is able to procedurally generate game levels that befit the player's skill.**

Keywords—Angry Birds; Genetic Algorithm; Procedural Content Generation; Game Levels

I. Introduction

Recently, game development has become more and more large-scale and complicated, in face of surging expectations of players. The cost and burden of game development have therefore reached unprecedented height. As a technology that can produce large-scale games with minimal manpower, procedural content generation (PCG) is gaining traction nowadays. PCG creates game contents automatically instead of manually. Evolution algorithm such as genetic algorithm (GA) is proposed as one of the techniques applicable in PCG. If the game's difficulty and player's skills match, the player will enjoy the game content more than otherwise [1]. In this paper[1], we propose generation of game levels that match the player's skills.

II. Angry Birds

Angry Birds is an action puzzle game franchise created by Finnish computer game developer Rovio Entertainment. It primarily involves shooting birds into pigs' fortress. Figure 1. is an example screenshot of Angry Birds. In this research, we used an Angry Birds clone developed by Lucas Ferreira [2], available for download via GitHub.

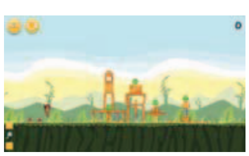

Fig. 1. A screenshot of Angry Birds.

[1] This work was supported in part by JSPS KAKENHI Grant Number 15H02939.

III. Level generation using Genetic Algorithm

GA is applied to regulate game difficulty. GA is an algorithm that imitates organic evolution. GA's mechanisms include evaluation of the fitness of individuals, selection according to the fitness, crossover and mutation. Based on the previous work by Ferreira and Toledo [2], this paper shows that procedural generation of levels that adapt to players' skills is possible by improving the fitness function and regulating the parameter of fitness function according to the game results played by the player so far.

A. Improvement of fitness function

The fitness function of individual *ind* proposed by Ferreira and Toledo is (1).

$$\min \ f_{ind} = \frac{1}{n}\sum_{i=0}^{n-1} v_i + \frac{\sqrt{(b-B)^2}}{M_b - B} + \frac{1}{1+p}, \qquad (1)$$

where the value $v_i\,(0 \le v_i \le 1)$ is the average magnitude of the velocity vector of element (block or pig) i , n is the total number of elements ($n = b + p$), b and p are the initial numbers of blocks and pigs in the game level of interest, respectively. The parameter B defines the desired number of blocks. In addition, M_b is the maximum number of blocks that a level can have.

GA then searches for the level that minimizes this fitness function. The first term of (1) evaluates the stability of the blocks and pigs in the level through their average velocity during the game. The second term is the distance between the desired amount of blocks B and the total number of blocks b in the level, normalized to 0-to-1 range. Due to the third term, the more pigs, the lower is the fitness. However, it is possible that the generated level has too many pigs, rendering a game that is unwinnable.

We change the third term in (1) to create levels that adapt to the player's skills automatically. To tackle this problem, a new fitness function (2) is proposed.

$$\min \ f_{ind} = \frac{1}{n}\sum_{i=0}^{n-1} v_i + \frac{\sqrt{(b-B)^2}}{M_b - B} + \tanh\left(\frac{|P-p|}{2}\right), \quad (2)$$

where the newly introduced parameter P defines the desired number of pigs. The new third term is the distance between the desired number of pigs and its actual initial number in the level, also normalized to 0-to-1 range.

B. Parameter regulation according to the results of playing

Now, we propose a mechanism for automatic creation of levels that adapt to the player's skills. This is done by regulating P according to the results of the player having played the precious levels that were created automatically. The player then plays the resulting generated level after evolution.

For this mechanism, we use k^{n+1}, the estimated number of pigs that the player can knock down in the $n+1$ th game, in (3).

$$k^{n+1} = a_{init}^{n+1} \frac{p^n{}_{init} - p^n{}_{left}}{a^n{}_{init} - a^n{}_{left}}, \quad (3)$$

where a_{init}^n is the number of birds given to the player in the beginning of the n th game, a_{left}^n is the number of birds left at the end of the n th game, p_{init}^n and p_{left}^n are the number of pigs in that level in the beginning and end of the n th game, respectively. In other words, k^{n+1} is estimated by multiplying a_{init}^{n+1} with the performance in the previous game. Thereby, P_{n+1}, the ideal number of pigs in the $n+1$ th game, can be predicted by (4).

$$P_{n+1} = (1-\alpha)k^{n+1} + \alpha P_n, \quad (4)$$

where α is a weight parameter. Note that P_{n+1} is calculated by summing the weighed values of k^{n+1} and P_n, the previous value. Moreover, B_{n+1}, the ideal number of blocks in the $n+1$ th game, can be given by (5).

$$B_{n+1} = \beta P_{n+1}, \quad (5)$$

where β is another parameter. This expression prevents situations where there are too many or little blocks in comparison with pigs.

IV. Experiment

We conducted an experiment with 12 players (9 males, 3 females with the age between 18 and 25 years old) to verify whether adjusting the parameters according to the player's gameplay results can generate game levels that adapt to their skills.

A. Methodology

The GA parameters were set empirically, i.e., the population size was set to 20 with the maximum number of generations of 1000, the tournament size of 2, the mutation rate of 5% and the crossover rate of 95%. In addition, the initial number of birds each game a_{init}^n was set to 3, P_0 = 5, α = 0.5, β = 10, and M_b = 100.

First, for each player, evolution based on the fitness function with the newly regulated parameters is carried out. The level of the individual whose fitness is the best is shown to the player for his or her to play. This process repeats 10 times. The value of P_{10} is then used in the following testing process. For testing, the player alternatively plays 5 levels created using their P_{10} (P_{player}) and 5 levels created using the average of P_{10} for all players ($P_{average}$).

B. Experimental results

The players were separated into two groups based on whether their P_{player} was higher than $P_{average}$. The average number of retries needed to clear the levels created using P_{player} and $P_{average}$ for each group is shown in Table I.

TABLE I. Average Number of Retries for each User group

		P_{player}	$P_{average}$
Player Group	$P_{player} \geq P_{average}$	0.25	0.25
	$P_{player} < P_{average}$	0.18	0.9

Table I shows the average number of retires to clear a given level is the same for both P_{player} and $P_{average}$ levels in the higher-performance group. For the $P_{average}$ levels, the average number of retries is much higher than that of the P_{player} levels in the lower-performance group.

V. Conclusions and Future Work

This paper proposed a method that automatically creates game levels for Angry Birds using GA. By adjusting the parameters of the genetic algorithm according to the player's gameplay results, our method can generate game levels that adapt to the player's skills. The result showed that the levels generated by the proposed methods can be cleared by all players; especially for those who cannot play Angry Birds well, it was less difficult for them to play with such levels. In future work, we will consider not only the number of pigs but also the kind of blocks in the fitness function.

References.

[1] Jenova Chen, "Flow in games (and everything else)," Communications of the ACM, Vol. 50, No. 4, pp. 31-34, 2007.

[2] Ferreira, Lucas and Toledo, Claudio, "A search-based approach for generating Angry Birds levels." Proc. of the 2014 IEEE Conference on Computational Intelligence and Games (CIG 2014), Dortmund, Germany, pp. 1-8, Aug. 26-29, 2014.