Interactive Evolutionary Programming for Mobile Games Rules Generation

Ong Jia Hui, Jason Teo, and Chin Kim On
Evolutionary Computing Laboratory
School of Engineering and Information Technology
Universiti Malaysia Sabah, Jalan UMS, 88400
Sabah, Malaysia
ongjh87@gmail.com, jtwteo@ums.edu.my, kimonchin@gmail.com

Abstract— **In this paper we present the possibilities of implementing evolution algorithm on a mobile platform. To inspect such possibilities we have created a Pac-man like mobile game that is implemented together with evolutionary programming. Here the interest does not only lie on performing the evolution algorithm *on the mobile platform* but also to examine the possibilities in implementing interactive evolution and create outputs that are based on the user preferences via reactive feedbacks, thereby creating a game design that is based upon the users' preferences. By explicitly examining the interactive evolution algorithm on a new testing platform, we have successfully created game rules that are based on the users' playing preferences.**

Keywords-component; *mobile platform, Android, Evolutionary Programming (EP), game design, Interactive Evolution Algorithm (IEA)*

I. INTRODUCTION

The usage of Smartphones is increasing rapidly and is expected to be the majority of phones used in U.S by 2011 [1]. Smartphones can provide a lot of functionality and yet statistics have shown that more time were spent on mobile gaming, thus it shows that the mobile gaming industry has been increasing tremendously in recent years [2]. iPhone operating systems (iOS) has been a major player in the Smartphone industry but the surge of Android with its open features might enable it to overtake the iOS as the major leading operating systems for Smartphone in 2011 [3]. This has opened an opportunity for us to explore the possibility to introduce Evolutionary Algorithms (EAs) to the mobile platform.

Evolutionary Algorithms (EAs) are a class of algorithms inspired from natural evolution that is used to help in solving hard computational problems. There are different classes of EAs like Genetic Algorithms (GA), Evolution Strategies (ES), Evolution Programming (EP), and Genetic Programming (GP). They differ from each other but operating on the same concepts and component of evolution such as populations, generations, and genetic operators. Interactive Evolution (IE) is a branch of EA as well, but in IE human decision is used to replace the normal EA fitness function [4]. Interactive Evolution Algorithm has been used in graphic art [5], music [6], and games [7].

In this paper, we created a Pac-man-like game in the Android operating system platform. EP is used for this research and we intend to test the combination of IEA together with EP in a mobile platform environment that has yet to be done in any study.

The organization of this paper is as follow. Section II presents the methods for this study, a detailed discussion on the development of the game, IEA and EP. Next section describes the experiment setup. Section IV discusses the experimental results. Conclusions and future works are given in the last section.

II. METHODS

Recently, several attempts have been made to create automatic content generations in games. With some work ranging from creating automatic platform game that generates it owns platform levels that are similar to the famous Mario game [8], while some studies have been implemented the automatic content generation in RTS to generate a competitive maps [10]. Togelius and Schmidhuber have recently created a Pac-man game-like that has no rules fixed to the game. The rules of the game were a part of the automatic generation content [9]. Based on this idea, we have created our own test-bed to examine our theory on mobile platform

A. Game

A game is created in Android platform that fit in a HVGA mobile display. The dimension of the game is 430 x 320 pixels. The game content consists of the following:

- Red rounded elements
- Blue rounded elements
- Yellow rounded elements
- Green rounded elements
- Cyan rounded elements
- 30 x 30 pixels white square

White border has been place in the game and serve as a container in the game. 40 white square is place in the game as walls that will restrict the elements movement. Yellow

elements will act as an element for human players. Others elements movement is shown below:

- Red and Blue static
- Green has horizontal movement only
- Cyan has vertical movement only
- Yellow (human player) can move in any directions

The position of Yellow elements is fixed in the middle of the screen. Other color elements position will be randomly place. Fig 1. shows the wall position and elements position in the screen.

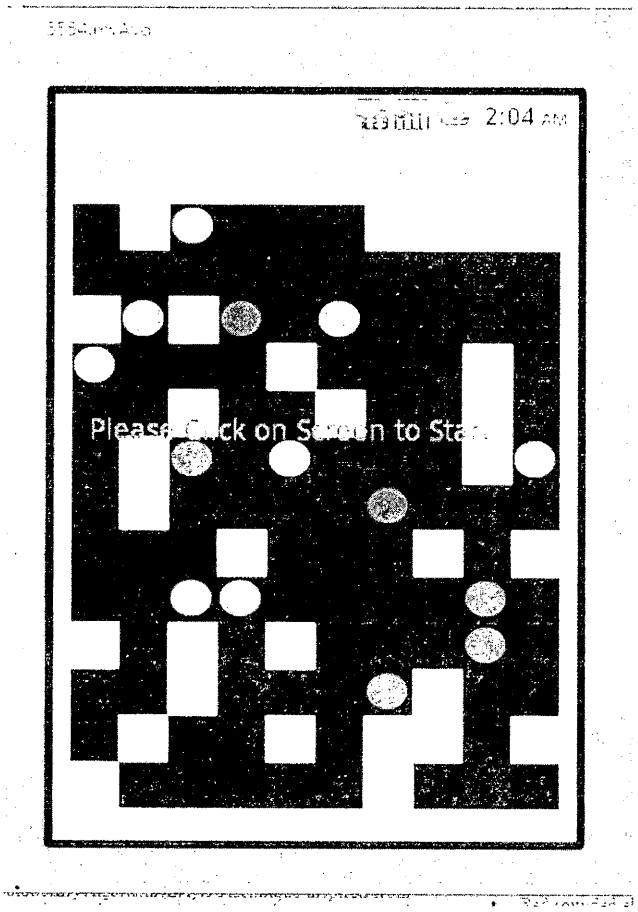

Figure 1. Game elements and walls position.

The wall position will be fixed for this research. There are two important components for the game which is the collision effect of each element and the score for each collision that happens. Table I shows the possible collision effect between elements.

TABLE I. COLLISION EFFECT

Elements	Yellow	Red	Blue	Green	Cyan
Yellow		C1	C2	C3	C4
Red	C1			C5	C6
Blue	C2			C7	C8
Green	C3	C5	C7		C9
Cyan	C4	C6	C8	C9	

C1: collision effect between Yellow and Red element
C2: collision effect between Yellow and Blue element
C3: collision effect between Yellow and Green element
C4: collision effect between Yellow and Cyan element
C5: collision effect between Green and Red element
C6: collision effect between Cyan and Red element
C7: collision effect between Green and Blue element
C8: collision effect between Cyan and Blue element
C9: collision effect between Green and Cyan element

Possible Effect:
0 = No Effect
1= Death (deleting the collided elements)
2= Teleports (elements back to its original position)

Due to the static movement of red and blue rounded elements the collision between red and blue elements is taken off.

TABLE II. SCORE EFFECT

Collision	Score
C1	S1
C2	S2
C3	S3
C4	S4
C5	S5
C6	S6
C7	S7
C8	S8
C9	S9

B. Evolutionary Algorithm

Evolutionary programming is applied in the study. The collision and score effects are used as the representation for the chromosomes. The numbers of possible elements for each color are ranged from 1 to 5 which mean there are at least 1 element for each color and maximum 5 elements for each color. These will be included as representation as well one more representation that will be used as representation is the winning score and losing score both range from 1 to maximum number of available elements present in current game. The size of the populations is $\mu + \lambda$. By using $\mu + \lambda$ populations it helps to decrease human fatigue during evaluation [4]. Below is the procedure of this

1.0 Start
2.0 Random initialize parent with the phenotype value for chromosome 0 to 17 is (0-2) while chromosome 18 to 21 will be (0-5) and the last two 22 and 23 the range of phenotype value is (0 to max number of elements). An example of chromosome encoding as shown below.

C1	C2	C3	C4	C5	C6	C7	C8	C9

Chromosome values from 0 to 8, encoding for collision effect

S1	S2	S3	S4	S5	S6	S7	S8	S9

Chromosome values from 9 to 17, encoding for score effect

R	B	G	C	W	L

Chromosome values from 18 to 21 are the encoding for the number of elements of red, blue, green and cyan. While chromosome value 22 to 23 is the winning and losing score for the game.

3.0 Load Parent into game and evaluate
4.0 Create offspring from parents
5.0 Load offspring into game and evaluate
6.0 Compare parent and offspring if offspring score better then parent, offspring will be parent for next generations
7.0 Step 3.0 to 6.0 will be repeated for 20 rounds of game.

C. *Interactive Evolution Algorithm*

Reactive and proactive interactions are two different branch of IEA. Reactive feedback is the algorithm that will request a feedback from user or the users can optionally intervene the autonomously running algorithm [11]. While proactive feedback, user can pause an algorithm that are in stagnation stage, alter some of the parameters and allow it to continue on it process [11].

The IEA that we have used in this paper is the reactive feedback method [11]. After a human player has completed a single game generations, user is required to assign a score to game. The score value range from 0 – 5 [4] (0- not satisfied with the current rules, 5 – likes the rules and would like to keep them)

III. Experiment Setup

The experiment has been conducted with the help of 21 students gathered from a faculty. Each student is brief on how the games work and method of playing the game before hand to eliminate confusion during game play that might affect the result obtains. Students need to go through the following steps in order to complete the whole run.

1.0 Start
2.0 A games rules is loaded into the game environment
3.0 Student played the game according to the current rules generated
  3.1 Games end either the player win or lose. In a case where the game cannot be ended (winning or losing score cannot be achieved) users can terminated the game.
  3.2 Users assign score for current games rules
  3.3 Next game rules generated and loaded into the game
4.0 Step 3.0 to 3.3 will be repeated for 20 rounds of games. Which mean each student will have to play 20 rounds of game since the number of generations has been set to 20.

Mutation rate is set at 0.1 for this experiment. $\mu + \lambda$ population size allows us to only choose the best offspring that score better compare to the current parent to be seeded as parent for next generations. This experiment has been conducted in a mobile simulator that allows us to access the same environment of a mobile device HVGA (320x480, medium density, normal screen) Fig 2 shows the screen shots of the simulator used.

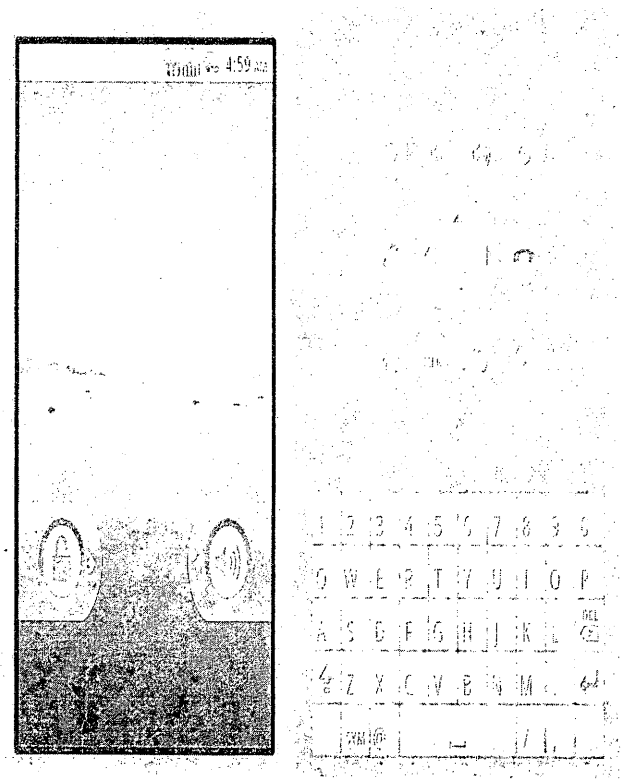

Figure 2. Android simulator screenshot.

Table III shows the summary of the experiment setup that has been used for this paper.

TABLE III. Experiment Summaries

Populations Size	$\mu + \lambda$
No. of generations	20
Mutation rate	0.1
Selections method	Survivor Selection
Fitness rating	0 – 5

IV. RESULTS

The results that we obtain from 21 users are shown in Table IV. The highest, lowest and average score of each runs is calculated. About 57% of users have given a rating of 5 at least once in the 20 generations. 19% of user gave the highest rating of 4, while 14% of users gave 3 as the highest rating. There are only 4% of user each that gave a rating of 2 and 1. The highest average rating given in the 21 students group is 4.619.

TABLE IV. RUNS RESULTS

Runs	Highest Score	Lowest Score	Average Score
1	3	1	1.857143
2	5	1	2.714286
3	1	1	3.190476
4	2	1	1.285714
5	5	1	2.238095
6	4	1	2.619048
7	3	1	2.095238
8	5	1	2.52381
9	5	1	2.619048
10	5	1	2.666667
11	4	1	1.904762
12	3	1	2.047619
13	5	1	3.428571
14	5	1	3.52381
15	5	1	3.333333
16	4	1	3.190476
17	4	1	3.142857
18	5	1	2.238095
19	5	1	3.952381
20	5	1	2.952381
21	5	2	4.619048

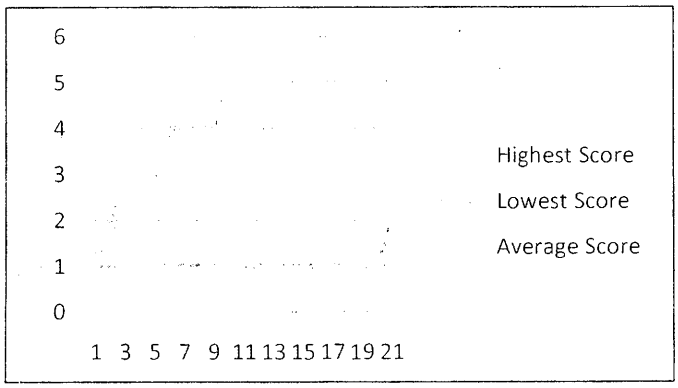

Figure 3. Result graph

Figure 3 shows the result in graph form, the average score shown in the graph reflects that almost 71 % of the average score lies between 2 and 3 region. We can conclude that most of the rules generated were in a satisfactory range for most of the players. Only 1 out of 21 runs that obtain rating 1 which we consider as bad rules games generated throughout the 20 generations.

During the experiment, one of the observations that we obtain is that if the initialization of parent is in the bad region, the following offspring will not get any better. We postulate that it is one of the problems that arise when $\mu + \lambda$ is used.

Human error will occur although not often but there are still possibilities of this event happen. This could affect the result obtain thus, user should be brief on what should and should not be done to minimize such error from happening. Fig. 4 and Fig. 5 is an example of rules generated that obtain rating 5. Fig. 6 is an example for rules generated that obtain a rating 4. Fig. 7 shows rules generated that obtain rating 3, while Fig. 8 and 9 is rules generated that attained a rating of 2 and 1.

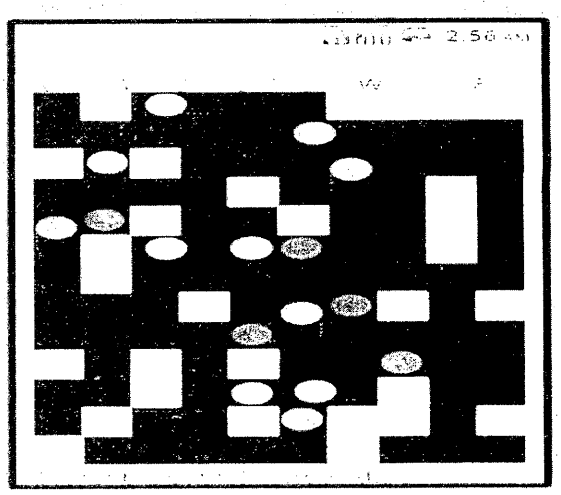

Figure 4. Chromosome 0 2 2 2 2 0 2 1 2 2 1 1 1 2 2 0 2 0 5 5 1 4 1 3 8

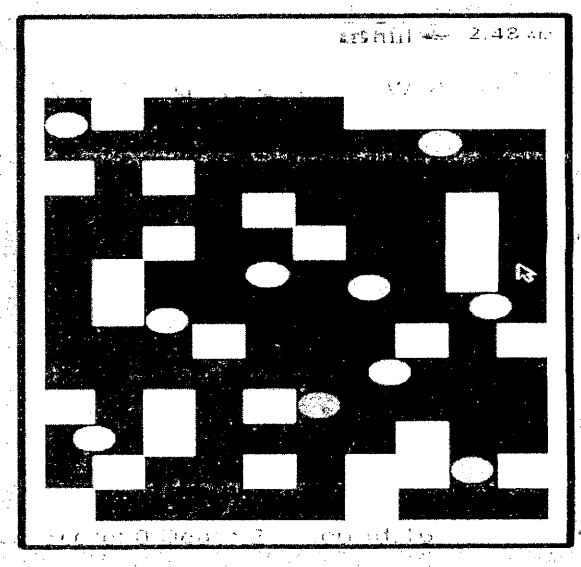

Figure 5 Chromosome 2 1 0 1 0 0 0 1 0 2 1 0 1 0 1 0 1 1 3 5 1 0 4 2

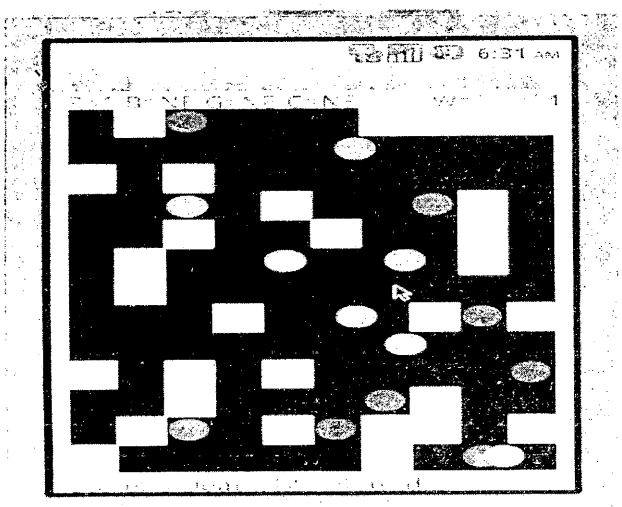

Figure 6. Chromosome 1 0 2 0 2 0 1 1 1 1 0 0 1 0 0 0 1 0 2 5 3 5 1 1 1

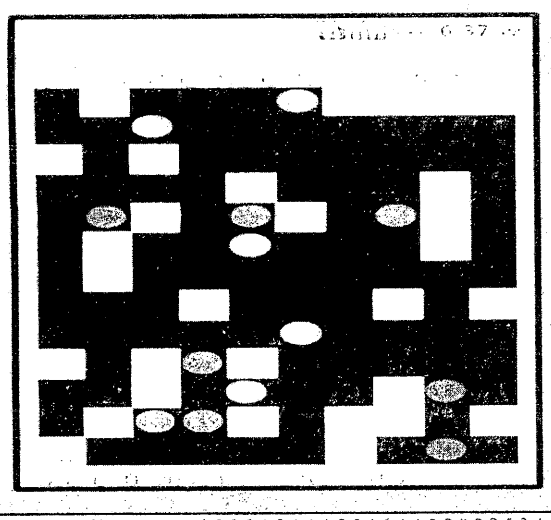

Figure 7. Chromosome 0 2 0 0 1 2 1 1 1 2 2 1 0 1 1 2 2 0 2 2 5 3 4 1 1

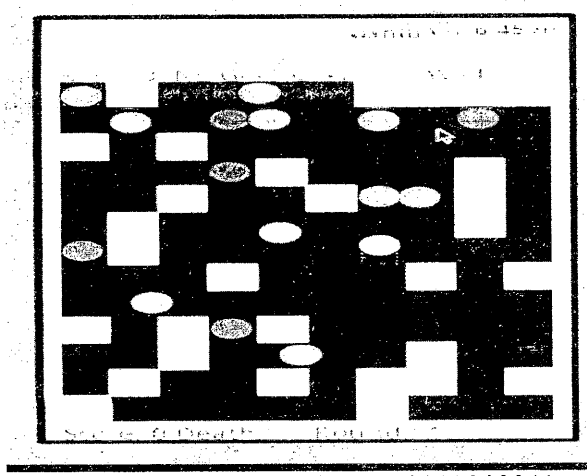

Figure 8. Chromosome 2 0 2 2 2 2 2 0 2 0 0 2 2 1 2 0 1 0 3 0 3 2 4 1

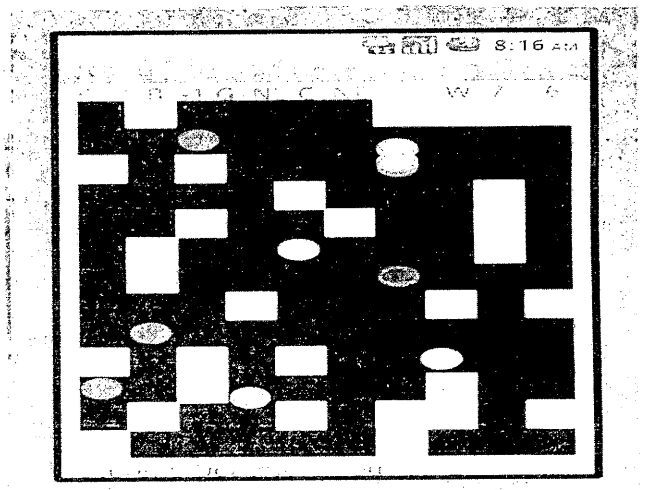

Figure 9. Chromosome 1 2 0 1 0 1 0 0 0 2 2 0 0 2 0 0 0 0 1 1 4 1 7 6

During the experiment process, one of the observations that we obtain is that if the initialization of parent is in the bad region, the following offspring will not get any better. We figure that it is one of the problem arise when $\mu + \lambda$ is used.

Human error will occur although not often but there are still possibilities of this event happen. This could affect the result obtain thus, user should be brief on what should and should not be done to minimize such error from happening

V. Conclusion Future Work

Implementation of IEA in mobile platform games is introduced in this research. The proposed algorithms were tested by users and the experimental results proved that implementation of EA into mobile platform can be done successfully.

From the results that we obtained, future work should be done on finding a suitable population size that fits in with IEA taking considerations on the human fatigue, large populations size would probably exhausted users and might cause more errors to happen. While population size being too small will affect the generated rules becoming stuck in local optimal, using larger and more optimum population sizes may aid in getting better results.

Altering the mutation rate and conducting a set of experiment on different ranges of mutation rates can be considered for future work as this might provide more insights into generating more favorable results. There is one more area that we need to look into for future work, and that is combining the reactive and proactive feedback that can enable users to either halt a process, change the parameter that he or she feels possible or the user can just give a rating at the end of the game.

Acknowledgment

This research is funded by the FRGS Research project FRGS0213-TK-2010 granted by the Ministry of Science, Technology and Innovation, Malaysia.

REFERENCES

[1] J. Rocha. "The Droid: Is this the Smartphone Consumers are Looking For?." Nielsenwire. 11 November 2009. http://blog.nielsen.com/nielsenwire/consumer/the-droid-is-this-the-smartphone-consumers-are-looking-for/

[2] S. Jones, "Rise of Mobile Phone Gaming Hurts the Old Guard," May 2004. http://news.brothersoft.com/rise-of-mobile-phone-gaming-hurts-the-old-guard-10028.html

[3] B.Proffitt. "Open Android-For Better and For Worse." in Spectrum IEEE volume 48 number 5 international page 22-23

[4] H.Takagi. "Interactive Evolutionary Computation:Fusion of the Capabilities of EC Optimization and Human Evaluation." in Proc. of the IEEE 2001. 22 pages

[5] P.J. Angeline, "Evolving fractal movies." in 1st Annual Conference on Generic Programming Standford. Ca. USA pp 503-511 1996

[6] Horowitz."Generating rhythms with genetic algorithms." International Computer Music Conference pp 142-143 1994

[7] E.J.Hastings, K.G.Ratan. . and K. O. Stanley. "Evolving Content in the Galactic Arms Race Video Game." in Proc. of the IEEE Symposium on Computational Intelligence and Games (CIG09). Piscataway. NJ:IEEF 2009

[8] K.Compton. M.Mateas. "Procedural Level Design for platform Games." in Proc. of the Second Artificial Intelligence and Interactive Digital Entertainment International Conference (AIIDE) Marina del Rey. 2006

[9] J. Togelius and J. Schmidhuber. "An experiment in automatic game design."in Proc. of the IEEE Symposium on Computational Intelligence and Games. 2008.

[10] J. Togelius. M. Preuss. and Yannakakis.G.N." in Proc. of the 2010 Workshop on Procedural Content Generation in Games 2010

[11] R.Breukelaar. M.Emmerich. T.B ck. "On Interactive Evolution Strategies." EvoWorkshops. pp530-541 2006