ISSN: 2454-9940

# INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org
Vol 14, Issuse2, April 2022

# Combining Genetic Algorithms with Other Rule-Based Systems to Boost Platform Game Efficiency

PRATHYUSHA TUMMEPALLY[1] , SAHITHI NAGUBANDI [2] , NAGALAXMI GUJJULA[3] , NIKITHA ANTHATI [4]
M.SHAILAJA[5]

**UG students[1,2,3,4,]  Associate Professor[5]**

## Abstract

***The Genetic Algorithm has been tried out in a number of different games with mixed results. The plat forming genre includes Geometry Friends, among many others. This study demonstrates how implementing GA may boost performance in a platform game. Using the game Geometry Friends, we aim to find the optimal values for the rule-based systems. In order to not solve the issue, we do tests on it. Reliant on a set of rules and regulations. We apply GA to each subset of experimental situations. This allowed us to determine the best values for the parameters. Tuning the parameters of a rule-based system using GA is more successful than using a rule-based system alone to boost performance.***

## 1. Introduction

One of the most common kinds of video games is the plat former. Typical elements of platform games include both platforms and hazards. The player has the ability to leap over barriers and collect various objects. Super Mario, Donkey Kong, and many more are all examples of well-known platform games. One crucial aspect of this plat former is the requirement for precise control. When the player must make a calculated decision, such as where to go or how high to leap to escape an obstacle, the stakes are high. One of the most crucial parts of a platform game is the player's ability to maneuver with finesse. In this article, we'll use a game called Geometry Friends to illustrate our method. The 2013 IEEE Conference on Computational Intelligence in Games (CIG) includes a competition called "Geometry Friends." It's a video game where you and a friend work together. It's a 2D platform puzzle game with realistic physics (including gravity and friction). The objective of the game is to quickly amass a diamond set [1]. Rule-based systems, which use a wide variety of characteristics to model character behavior, power the AI-agent in Geometry Friends. In this research, we apply the Genetic Algorithm (GA) to the Geometry game in order to fine-tune its settings. In other contexts, GA is utilized to address a wide range of issues [2].

In fact, several fruitful GA-based approaches have been investigated [2, 3]. Using GA, Chishyan Liaw et al. [4] implemented team evolution in a first-person shooter. In a platform game, Ken Hasegawa et al. [5] suggested utilizing GA to Determine meta-level action in response to surroundings. In this study, we want to zero in on the specifics of how game-based rule-based systems function. We begin by understanding the limitations of rule-based systems and pointing out the unclear parts of the system that they cannot resolve. Then we put into practice the game's nuanced mobility. In order to better absorb information, we conducted an experiment to compare different scenarios. Our experiment's goal is to enhance the game play of a platform game by using GA. The following is the outline for this paper. Session 2 will be an introductory talk to Geometry Pals. Our experiments' genetic algorithm is presented in session 3. Session 4 is when we do the task experiment and share the findings. In the last session, we draw conclusions about our experiment and discuss potential next steps.

## 2. Geometry Friends

### Score system

A predetermined set of 10 levels will be used to test the AI players. The score is calculated by adding up the number of collectibles players have found in each level and how long it took them to finish the level. In order to reduce the number of submissions, we will test the AI players in each level ten times. Each level's final score will be based on the mean of those ten runs. The A run's score will be determined as follows on each tier:

$$Score = (N_{get} * V_{get}) + V_{bonus} * (maxTime - playTime)/maxTime \quad (1)$$

Within bounds, both players may influence its movement and form. The ball may hop and change size, and both characters can move horizontally. It's possible to alter the square's orientation, making it horizontal or vertical. In Fig.1, we can see the various characters in action. There are real-world physics at play, such as gravity, friction, velocity, and acceleration, in this game. There are many similarities between this game and the current smash hit Angry Birds. The player doesn't have direct control over nuanced motion. When the player inputs a command, the character does the appropriate action. As an example, Square's acceleration and speed increase as the player repeatedly presses the right command. Then, effective regulation of that kind of conduct is essential. The definition of expected behavior is critical in rule-based systems. Choosing all rules on one's own, however, is quite challenging. In this study, we use the GA to find solutions to the issues at hand.

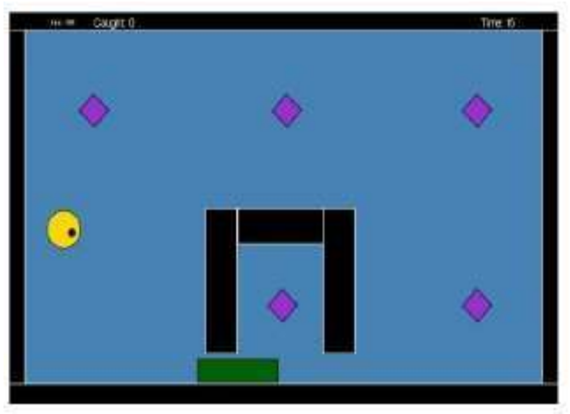


*Fig. 1. Example of Geometry Friends map*

## 3. Hybrid of Rule-based Systems and Genetic Algorithms Genetic Algorithm (GA)

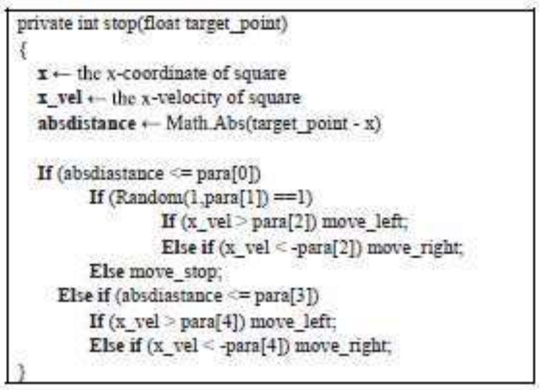
Referring to Fig.3, in this study, we use a traditional Genetic Algorithm [6]. Generations are the basic unit of time in population studies. To begin, a random pool of potential parents (P) is drawn from. Each population's parameters are set to a random value between the lowest and maximum values in the starting parents (P) that are specified. Next, we use a game simulation to assess the population's performance based on the scoring system described in Section 2.1. Current population members serve as the parents (P). Mutation and genetic crossover allow parents to have offspring (p). It is important that the value of, which represents a high population answer ratio, be passed down from one generation to the next. The iterations of GA may go on forever. This allows us to get the best possible gaming settings. Parameters like the agent's velocity, break point, direction, and etc. must be defined if it is to reach the game's objective. However, it is difficult to specify all parameters precisely. Therefore, we use GA to look for good estimates for these parameters. Using these values, a population may be generated.

### Rule-based systems (RBS)

Multiple components make to a rule-based system (RBS). Through RBS, the game agent is able to

ISSN2454-9940www.ijsem.org
Vol 14, Issuse2, April 2022

acquire a diamond, the game's ultimate objective. In this article, we analyze the game's most critical issue, the halt () method There are several functions, including halt (). When you use the stop () method, your character's velocity slows down gradually in the same It's quite close to the mark. When it reaches the intended destination, the square agent will stop. Using GA, we may fine-tune the values of Para [0] through Para [4] in Fig.2. When tampered with by humans, they perform poorly in games. The use of RBS alone enables the creation of AI. However, expecting good results with merely one RBS agent is really challenging.



```
private int stop(float target_point)
{
  x ← the x-coordinate of square
  x_vel ← the x-velocity of square
  absdistance ← Math.Abs(target_point - x)

  If (absdiastance <= para[0])
        If (Random(1,para[1]) ==1)
              If (x_vel > para[2]) move_left;
              Else if (x_vel < -para[2]) move_right;
        Else move_stop;
    Else if (absdiastance <= para[3])
        If (x_vel > para[4]) move_left;
        Else if (x_vel < -para[4]) move_right;
}
```

**Fig. 2. Pseudo code of rules**

## Optimization of RBS using GA

A combination of RBS and GA is required. RBS are the backbone of our infrastructure. Here, we use GA to the task of optimizing the primary RBS. In Fig.4 we see the GA simulation process shown on a per-iteration basis. Similarly, Para [0, # of parameters] is modified by iterative GA. In each cycle, this method runs 10 simulations. Eventually reach a level of fitness of 10. The players immediately entered the fitness function when the game concluded. Then we can learn everyone's fitness score. The goal of this process is to get RBS settings that have been optimized.

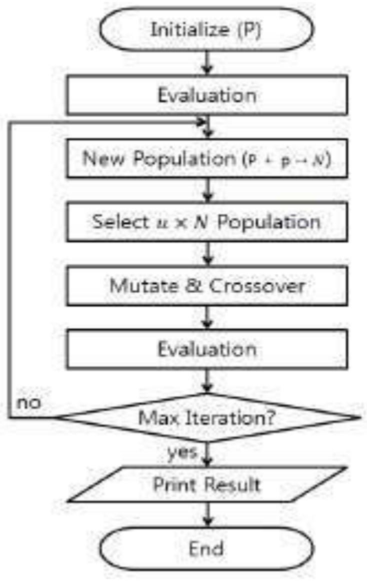


*Fig. 3. Flow diagram of Genetic Algorithm*

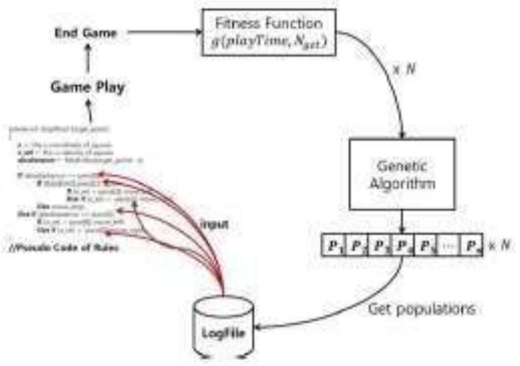


*Fig. 4. Simulation procedure per iteration*

## 4. Task Experiment

To test this hypothesis, we conduct experiments in which the issue is not solved (the map is not cleared perfectly in the game). We use GA to create and solve such problems. In addition, this investigation determines what kinds of cases benefit from parameter adjustment.

### The Laboratory Setting

The white box in Fig. 5 represents a green square, whereas the black box represents potential roadblocks. The dot seen in Fig.5 represents the exact center of the gap depicted there. Distances from the center to the edges are arbitrary. If the white box, which represents a character in our experiment, were to fall into the hole, the level would be considered cleared. However, it is not simple for the player to handle the sensitive movement of this game, making it a task that is tough to utilize RBS. If your goal point is, you may utilize the halt () method described in subsection 3.2. Five of these stages have already been unlocked for the tournament. We anticipate a satisfactory performance on around four of the five levels if the halt () function is implemented. To slow down, a baseline is used as a limit. The only way for a human to learn many parameters is via trial and error. It's quite inefficient, however. Therefore, GA allows us to get the best possible results in each given circumstance.

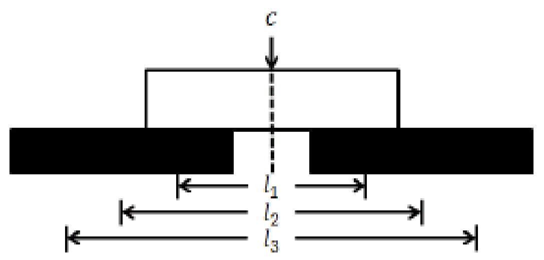


*Fig. 5. Map imaging 4.2. Experimental setup*

We conducted three separate trials, which we will refer to as A1, A2, and A3.

**Experiment Case Table 1**

| Case | # of parameters | Used the boundary ($l_k$) |
|---|---|---|
| *A1* | 4 | Only $l_1$ |
| *A2* | 6 | $l_1$ and $l_2$ |
| *A3* | 8 | $l_1$ and $l_2$ and $l_3$ |

This was GA's medical status: Maximum iterations are 50, and their values are 0.3 and 10. GA was simulated 500 times for each situation as can be seen in Fig.6, each potential candidate writes to the log file each time a generation of GA completes. Top 8~10 the log file is mined for potential candidates. The reason there aren't more than 10 people chosen is because the number of applicants who meet the minimum requirements is only 8 in case and only 9 in cases. Each finalist does 50 simulation runs. Each candidate's score from the simulation system is recorded. The best candidate is selected from the top 8–10 to determine the success rate of each. The best possible case parameters are the ones with the highest ranking. There is a chance that the GA's final candidate will not be the best one since it may have fallen into a local optimum. In order to determine which of the top 8–10 candidates is the most likely to "fall into the hole," we have made our selections. The most qualified individual will get the best results.

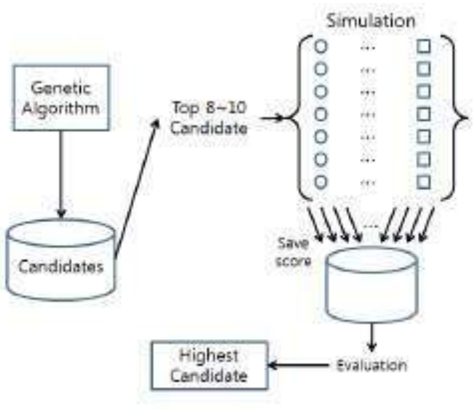


*Fig. 6. The schematic processing of experiment*

## Results

*Table 2. The success probability of cases*

| Case | Probability of last candidate | Probability of highest candidate |
|---|---|---|
| *A1* | 0.18 | 0.24 |
| *A2* | 0.6 | 0.8 |
| *A3* | 0.08 | 0.18 |

Table 2 shows that in every scenario, the last remaining contender performed poorly. They both accomplished above-average work in each scenario, but the other top pick ultimately won out. The difference between and was 62%. Based on these results, we know that GA can get a collection of optimum stop () values.

## 5. Conclusion and Future work

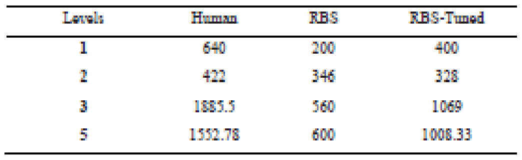
In this article, we show how to use GA to find the sweet spot for your game's settings. Table 3 displays the comparative findings between humans, RBS, and RBS-Tuned. Each level's score was measured using the game's scoring system five times. Since the stop () method modifies the levels, we choose 1, 2, and 3 as cutoffs for our scoring. Real-world game play is significantly enhanced with GA-based parameter adjustment. Offers a tremendous value at levels 3 and 5. Rule-based systems and RBS-Tuned systems are clearly distinct from one another, as discussed in Section 2. In the future, we plan to use this method to create even another plat former. We'll also look at

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org
Vol 14, Issuse2, April 2022

methods of rapidly iterating on improvements to performance.

***Table 3: Levels of Results***



| Levels | Human | RBS | RBS-Tuned |
|---|---|---|---|
| 1 | 640 | 200 | 400 |
| 2 | 422 | 346 | 328 |
| 3 | 1885.5 | 560 | 1069 |
| 5 | 1552.78 | 600 | 1008.33 |

## References

*[1] http://gaips.inesc-id.pt/geometryfriends/*

*[2] Hasegawa, K., Tanaka, N., Emoto, R., Sugihara, Y., Ngonphachanh, A., Ichino, J., and Hashiyama, T., Action selection for game play agents using genetic algorithms in platform game computational intelligence competitions, Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 17, no. 2; 2013, pp. 201-207.*

*[3] Goldberg, D. E., Genetic algorithms in search, optimization, and machine learning. Reading, Mass. : Addison-Wesley, 1989.*