

# Using A\* and Q-learning Algorithms to Implement the Task of Geometry Friends Single AI Track

Yen-Wen Lin, Li-Jia Wei and Tao-Hsing Chang

Department of Computer Sciences and Information Engineering  
National Kaohsiung University of Applied Sciences, Taiwan

## 1. Introduction

The task of Game "Geometry Friends" is to gain all of the diamonds on the map within certain period of time. On this bird's-eye view map, players are able to visualize full images and locations of the diamonds at the onset time. Hence, we may design an A\* algorithm agent to search for the shortest path in the fastest time, aims at Game 'Geometry Friends' to alter and optimize the agent. Eventually, make this optimized the agent to gain all of the diamonds within the fastest/shortest period of time. Moreover, there are many difficult pitfalls on the path generated by A\* algorithm. This paper employs Q-learning algorithm to learn how to through these pitfalls.

The contents of this paper are shown as follows. Section 2 is to point out the problems when we manipulate "agent" to gain the diamonds Section 3 is to introduce the method(s) we use to refine and optimize A\* algorithm agent. Section 4 explains how to apply Q-learning to the games.

## 2. Key Points to Accomplish Task

It is crucial and critical to arrange the order of a sequence for gaining diamonds. If the sequence is not with proper arrangement, players may delay the task, or even fail it. Figure 1(a) as example, there are six sequences to arrange the order of diamonds, but only one way, from left to right order, is the possible method to gain diamonds. Because of circle's limited motion, it's not reachable to gain the right-hand-side diamond directly. In this case, to gain the left-hand-side diamond and occupy the platform/position is prior to other diamonds. That's why arranging the order of diamonds' sequence is the key point of Geometry Friends task.

Figure 1(b) is the other example to draw the importance of arranging the order of diamonds' sequence. What if we plan to manipulate circle to gain the diamond between two black platforms first, and then move to the diamond on the left-up-corner. It's easy to gain the first diamond, but it's difficult to escape from the gap between two platforms and gain the second diamond.

In contrast, if we plan to manipulate circle to gain the diamond on the left-up-corner first, and then move to the diamond between two black platforms, it would be a better

choice rather than the previous method. Even though we spend time on the longer route, eventually it's a time-saving choice compare to escape from the gap. This paper aims at how to design an A\* algorithm agent with reasonable, time-saving, considered arrangement of diamonds sequence.

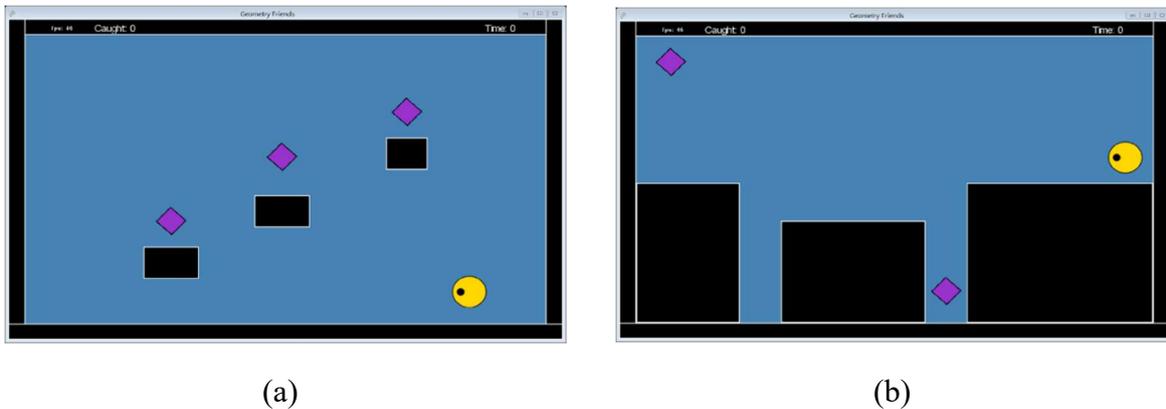


Fig. 1: Two Examples of Geometry Friend Task

### 3. The Use of A\* algorithm

A\* algorithm is the method to figure out the best solution. We apply A\* algorithm to figure out the best solution of arranging diamonds' sequence. Take "Rectangle Edition" as the example, the pseudo code is as follows.

```

Add START to OPEN list
while OPEN not empty
  get node n from OPEN that has the lowest f(n)
  if n is GOAL then return path
  move n to CLOSED
  for each n' = CanMove(n, direction)
    g(n') = g(n) + 1
    calculate h(n')
    if n' in CLOSED list and new n' is not better, continue
    remove any n' from OPEN and CLOSED
    add n as n''s parent
    add n' to OPEN
  end for
end while
if we get to here, then there is No Solution

```

where  $f(n) = g(n) + h(n)$ ;  $g(n)$  represents for the movement cost from the start point to node  $n$ ;  $g(n)$  is the movement cost from the start point to node  $n$ ;  $h(n)$  represents for  $D(n')$  plus the direct distance between the node and the nearest diamond;  $h(n) = D(n') +$  the direct distance between the node and the nearest diamond;  $D(n')$  represents for the amounts of searching towards to the specific direction. If we changed the searching direction, reset  $D(n')$  to zero.

#### 4. Q-learning

Q-learning is a type of algorithm in machine learning, and Q-learning is widely used in AI games to optimize agent for fitting particular environment. This paper applies A\* algorithm to arrange the best path for gaining diamonds. Then, we use Q-learning to figure out and adjust the alternative motions, in order to move along the best path. An agent designed by Q-learning would keep trying various possible movements over and over, to learn the way how to through pitfalls quickly. Finally, come up with the best solution.

Q-table is the core of Q-learning. Q-Table used in this paper train and adjust seven parameters, including the path and direction based on the results of A\* algorithm, possible movements, rectangle height, and so on.

#### 5. Experimental Results

This paper carries out the proposed agent on computers. Table 1 and 2 show the agent performs on circle and rectangle track, respectively. Although the computers the of operating the agent may be difference from that of 2013 winner, the preliminary results indicate the proposed agent works well.

Table 1. the results on circle track

Level	Level Photo	Runs Completed	Time in secs	Diamonds Collected
Jump It	Our agent	10	2(2)	10.0(20)
	2013 Winner	10	2(2)	11.2(20)
Jump it Twice	Our agent	10	3(3)	18.0(20)
	2013 Winner	10	3(3)	19.7(45)

Table 2. the results on rectangle track

Level	Level Photo	Runs Completed	Time in secs	Diamonds Collected
The Fall	My AI	5	1.5(2)	27.3(40)
	2013 Winner	10	2(2)	11.4(40)
Gravity Friends	My AI	10	2(2)	7.8(20)
	2013 Winner	7	1.4(2)	12.4(20)
Fall on the Other Side	My AI	5	2.3(3)	29.3(35)
	2013 Winner	0	1.8(3)	35()