Solving Geometry Friends using Monte-Carlo Tree Search with Directed Graph Representation

Hyun-Tae Kim, Du-Mim Yoon, and Kyung-Joong Kim*
Dept. of Computer Science and Engineering
Sejong University, Seoul, South Korea
kimhyun0416@sju.ac.kr, krad@hanmir.com, kimkj@sejong.ac.kr

***Abstract*— Geometry Friends is a platform game with two different objects (rectangle and circle). Players can control the geometries to collect all the diamonds in the level. It requires both of efficient path planning and careful objects control with physics law. It has been revealed that it is not easy to achieve human-level performance for the game. In our early work, we show that simple A* search can be used for the problem. Although it's useful, it sometimes falls into local optimum. In this work, we propose to use Monte-Carlo Tree Search (MCTS) to generate plans for the game object. Because the game levels are not suitable for the MCTS, we propose to represent them into a directed graph with special properties. Experimental results demonstrate the potential of the MCTS approach for the Geometry Friends game.**

Keywords*— Path-Finding; Monte-Carlo Tree Search; Physical Travelling Salesman Problem; Platform Game; Geometry Friends;

I. Introduction

Platform game is a popular video game genre that consists of platforms and obstacles to be used by the character to move and jump. The main objective of this platform game is that the character usually moves on platform and jumps to another platform in order to achieve some objectives (find an exit, kill enemies and collect bonus items). For example, the super Mario moves forward, backward and jump to clear each level on the platform. For several years, there have been competitions for the super Mario for playing, learning and human likeness.

Recently, a new platform AI competition has been introduced using "Geometry Friends" since 2013 (http://gaips.inesc-id.pt/geometryfriends/). The player can control rectangle and circle objects to collect diamonds on the level. The goal is to collect all the diamonds for a short time. The rectangle and circle objects have different capability. For example, rectangle objects can slide sideway and resize (no mass change). However, circle objects can roll sideway and resize (mass change). To solve complex problems, the two different objects need to cooperate. It has a physics engine to simulate friction, acceleration and gravity.

In our early work [1], we use a grid representation for the level of Geometry Friends with A* search to find the shortest paths. Although the implementation is straightforward, the agent sometimes falls into stuck because it is based on greedy search. As a result, it is necessary to implement additional exception handling and heuristics to avoid the case.

The player on the Geometry Friends game has to acquire the specified number of diamonds for each level as soon as possible. In terms of action planning of AI agent, obtaining a diamond in the game is considered as a path-finding problem. On the other hand, the problem of obtaining all diamonds can be seen as a physical travelling salesman problem (PTSP). To solve PTSP, There are already a lot of studies in game area [2]. In the game on this PTSP league, an agent is able to move in and all directions usually are used to solve PTSP. However, we presented that the agent can consider multi-objective problem as PTSP and solve this problem in the Geometry Friends platform game with physical engine.

We found that the Geometry Friends have common concerns with Physical Travelling Salesman Problem (PTSP) competition. In the PTSP, the agent should pass all the waypoints in the map. In has been known that the MCTS is successful for the PTSP [2]. In this work, we attempt to use the MCTS for the Geometry Friends. However, it is not straightforward to represent the levels of the game into a tree structure. We propose an algorithm to create a directed graph with shape-assigned edges from the platform game level.

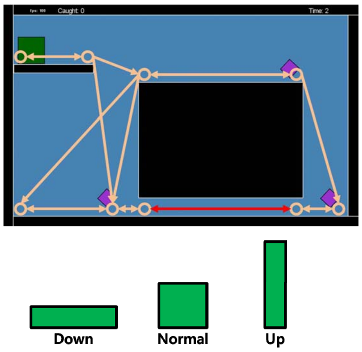
II. Proposed Method

It is not straightforward to apply the MCTS to the platform game. Because each level is not represented as a tree, it is necessary to estimate the possible paths for the object and create a tree-like structure for the level. Unlike other board and video games, the game also has several physical constraints (due to physics engine and platforms). Furthermore, the game object can resize the shape.

- Node: In the level, there are several platforms. For example, there are three platforms in the Fig. 1. Over the platform, the rectangle can slide sideway. The two ends (left and right) of each platform is a node. Also, the node is defined at the narrow alley. Finally, a node is added to the potential place where the rectangle will fall.
- Edge: The edge has a direction (one/two-way). If there is an arrow from node *A* to *B*, it means that the rectangle can move towards *B* from *A*. The rectangle object can resize the body. In this work, we consider only three types of typical change. "Normal" is square; "Down" is a flat shape and "Up" is a thin stick. The agent assigns one of the three shapes to the edge. For the narrow alley, the "Down" shape is the most appropriate and it is assigned for the edge. The assigned shape is used to simulate the node transition.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2013R1A2A2A01016589, 2010-0018950).
*Corresponding author

978-1-4799-3547-5/14/1.00 ©2014 IEEE



- Fig. 1. An example of directed graph for the *"fall on the other side level"* and three shapes of the rectangle

From the definition of the node and the edge (associated with one of three shapes), it is possible to construct a directed graph for the Geometry Friends map. The graph considers physical constraints (no "jump" and blocking by obstacles) for the object. The next step is to convert the graph into a tree for the MCTS (Fig. 2).

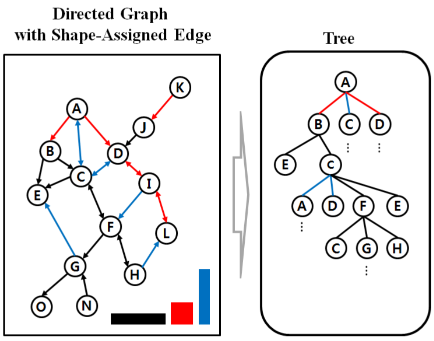


Fig. 2. The conversion of the directed graph into tree structure for MCTS (color means the type of shape)

The MCTS agent used the upper bound confidence bound (UCB) in the selection phase [3] and directed graph for the expansion phase. The heuristic function of the simulation phase is as follows:

$$(N_{Collect} + 0.01) * V_{Collect} * avg(V_{completed}) / len$$

,where $N_{Collect}$ is the number of diamonds collected and $avg(V_{Completed})$ means the average of completion bonus (given by 2013 competitions) and *len* denotes the distance between current node and root node.

III. Experimental Results

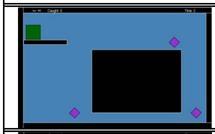
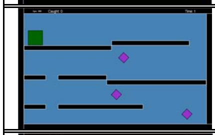
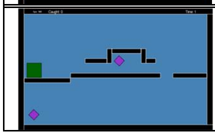
In this paper, we use the official scoring metric for the Geometry Friends competition. Each level has different difficulty, time limit (*maxTime*), completion bonus ($V_{Completed}$), and collectable unit value ($V_{Collect}$). To get high score, the player should collect as many as diamonds for a short time. Completion bonus is awarded if all diamonds are collected.

$$score = V_{Completed} * \frac{(maxTime - agentTime)}{maxTime} + (V_{Collect} * N_{Collect})$$

Although there are three tracks for the Geometry Friends competition, we focus on the "rectangle solo play" track. In this work, we compare the performance of A* (CIBOT submitted for 2013 competition), MCTS and human player on three levels (two middle and one hard difficulty) from the 2013 IEEE CIG Geometry Friends competition. Table 1 summarizes the statistics of results for the players. It shows that the MCTS outperforms the A* for the three levels. However, there is performance gap between human and the MCTS.

In case of *Fall on the Other Side* (1^{st} row), A* fails to get a diamond on top of obstacle because it collects the bottom diamond first. The rectangle cannot climb up the obstacle. However, MCTS selects a diamond on the obstacle as a first target. Although the diamond in the left bottom is the closest one to the rectangle's starting position, it's a local optimum if it is collected first.

TABLE I. Statistics of the Results (Average of 10 runs for A* and MCTS, Oracle is an IDEAL Performance)

Levels	Player	Diamonds Collected	Time (Sec)	Score
	A*	1.8	35	198
	MCTS	3	10.2	564
	Human	3	8	585
	Oracle	3	0	660
	A*	0.2	65	64
	MCTS	1	65	320
	Human	3	21	1095
	Oracle	3	0	1910
	A*	0	60	0
	MCTS	1.5	44.3	627
	Human	2	14	1095
	Oracle	2	0	1240

IV. Conclusions and Future Works

Geometry Friends platform game requires multi objective optimization (Time and Diamond Collection) to complete each level. In our 2013 entry, we introduce the use of A* search on grid representation of level. Although it's easy to implement, it is likely to be trapped into local optimum. In this work, we propose to use MCTS for the path planning of the rectangle object. Because the map is not represented as a tree structure, we propose to convert the map into a directed graph considering the physical constraints of game objects. Experimental results show that the MCTS with the directed graph representation outperforms the simple greedy approach on the three levels from the 2013 Geometry Friends (rectangle solo play track) competition.

In the Geometry Friends, there is a circle object which can "jump" and "resize (mass change)." It is different with the behavior of the rectangle object. There are additional tracks in the competition for the circle solo play and the cooperation of the two different game objects. It is necessary to use the MCTS for the different game object and the cooperation of them.

References

[1] H.-T. Kim and K.-J. Kim, "Hybrid of Rule-based Systems Using Genetic Algorithm to Improve Platform Game Performance," *Procedia Comp. Sci.*, vol. 24, pp. 114–120, 2013.

[2] D. Perez, E. J. Powley, D. Whitehouse, P. Rohlfshagen, S. Samothrakis, P. I. Cowling, and S. M. Lucas, "Solving the Physical Traveling Salesman Problem: Tree Search and Macro Actions," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 1, pp. 31–45, 2014.

[3] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, 2014.