The Geometry Friends Game AI Competition

Rui Prada* Phil Lopes†, João Catarino‡, João Quitério§, Francisco S. Melo¶
INESC-ID and Instituto Superior Técnico - Universidade de Lisboa
Av. Prof. Cavaco Silva, 2744-016 Porto Salvo, Portugal
* rui.prada@gaips.inesc-id.pt
† plopes@gaips.inesc-id.pt
‡ jcbpc@tecnico.ulisboa.pt
§ joaollquiterio@tecnico.ulisboa.pt
¶ fmelo@inesc-id.pt

***Abstract*—This paper describes a new game AI competition that engages participants in the creation of agents for a cooperative platform puzzle game. The agents face the challenge of acting in a dynamic environment, with friction and gravity, as they coordinate actions to solve cooperative puzzles. Hence, agents need to devise cooperative plans to solve the puzzles in the most efficient way and to coordinate their actions to perform joint actions in real-time. The core of the competition is the cooperative track that requires the development of two distinct agents, but we also include a single player track for participants that want to craft the basic skills of the agent without the complexity of cooperation. In particular, to cope with understanding the topology of a level in order to define a plan for solving the puzzle or to define mechanisms to act well in the physics-based game environment. We present the results of the 2014 competition held at IEEE Conference on Computational Intelligence in Games, in Dortmund, and discuss some future directions of research.**

I. Introduction

Competitions of Artificial Intelligence (AI) have been used frequently and successfully to foster the development of new and innovative research. Competitions, such as, RoboCup [1], the Supply Chain Trading Agent competition [2] or the Darpa Grand Challenge [3] motivate many researchers to work on difficult problems and provide, at the same time, a framework for common ground to contextualise and compare research.

Game AI competitions have proven fairly successful in the past as well. AI agents have been built to play classical games like Chess, Go or Poker [4]. There is even a competition of General Game Playing [5] where the agents play games they do not known beforehand. More recently competitions started to focus on AI for videogames (digital games) like Super Mario Bros. [6], Unreal Tournament [7], Racing Games [8] and Pac-Man [9], which brought the additional goal of creating AI to provide a good experience to the player besides solving the game. Hence, issues such as the believability of the agents gained importance.

We propose a new competition of Game AI[1] for a physics-based cooperative puzzle-platform game. The game, Geometry Friends [10], was developed with the purpose of providing a collaborative experience for two players using the Wiimote controllers. It was later extended with a map editor and a framework for the creation of artificial players to promote a single player experience. Because of its cooperative nature, the game promotes interesting AI challenges that are not often seen in game AI competitions. It provides challenges for cooperation at different levels. The agents need to define and execute cooperative plans to be successful. Planning and execution of the plan each come with their own set of challenges. First, agents need to identify opportunities for cooperation and to identify required cooperative steps; then they need to execute actions in a physics-based environment, with friction and gravity, which they often need to execute simultaneously in a precise coordinated way and in real time.

[1]Check the competition website for more details: `http://gaips.inesc-id.pt/geometryfriends`.

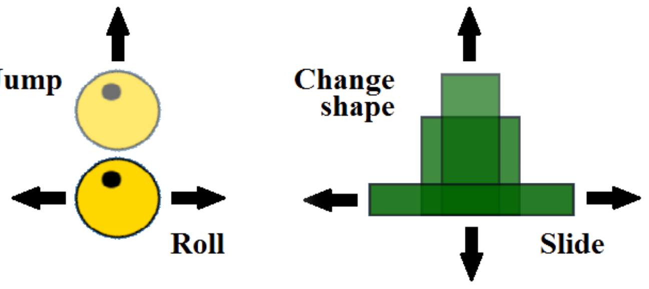

Fig. 1. The actions that each character (Circle and Rectangle) may perform. The Circle can roll and jump, the Rectangle may slide and morph.

In this paper we start by describing the Geometry Friends game with some detail and then discuss the main challenges for AI that are present in the game. Afterwards, we present some details of the API of the agent framework in order to give some insights regarding the kind of information the agents will access and need to process, followed by a section describing the competition rules. Then, we discuss the results of the 2014 Competition describing the main approaches used by the participants. We conclude the paper with a discussion of future research directions.

II. The Game: Geometry Friends

Geometry Friends is a cooperative skill-based puzzle game, where cooperation, problem-solving and time-based skills are crucial elements for solving the game's most complex levels. The game was created with two players in mind, although it was later extended to support single player levels. It is set in a two dimensional world with simulated physics. Each player controls one of two geometric figures, the yellow Circle and the green Rectangle, with distinct capabilities. The objective

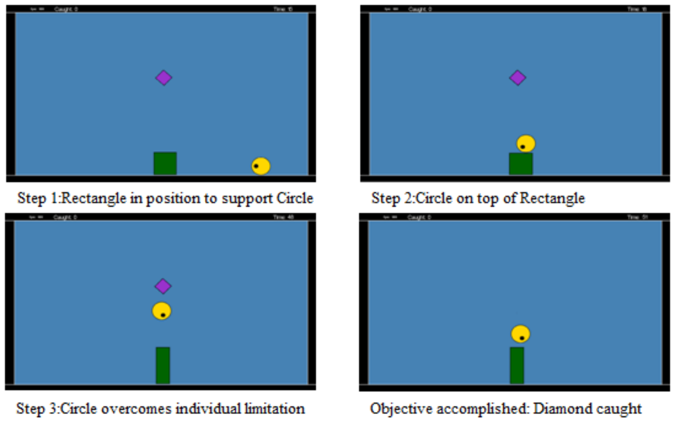
Step 1:Rectangle in position to support Circle
Step 2:Circle on top of Rectangle
Step 3:Circle overcomes individual limitation
Objective accomplished: Diamond caught

Fig. 2. The Circle may use the Rectangle, as a springboard, to reach higher places.

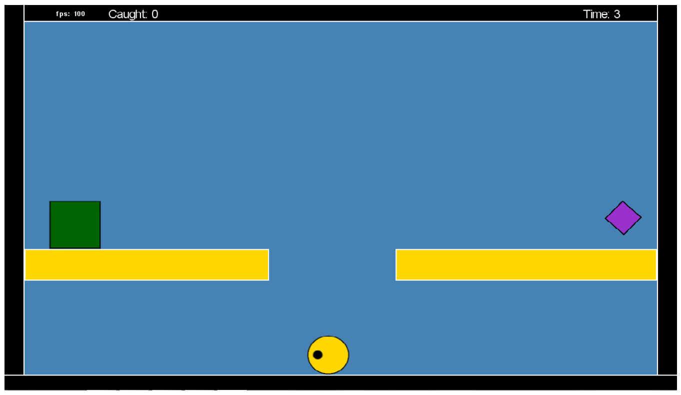

Fig. 3. A simple Geometry Friends level with coloured platforms.

of the game is to collect a set of diamonds scattered through the level and the score depends on the time spent to do so.

The Circle can jump and the Rectangle can change its shape shrinking and stretching its base while keeping the same area (see Figure 1). Both can move horizontally, by rolling and sliding, respectively, and are subject to gravity and friction. Both characters are able to collect diamonds. However, the distinct capabilities make certain parts of a level only reachable by one of the characters, other parts may be reachable by both, and, often, some are only reachable if the two characters cooperate. For example, the Circle can use the Rectangle as a springboard to reach high places that it cannot reach alone (see Figure 2).

In Geometry Friends, a level is defined by a set of walls and platforms that restrict movement, a set of diamonds positioned in the game world, and the initial position of the two characters. Some of the platforms are coloured (green or yellow). These restrict only the movement of one of the characters. The rule is that characters collide with anything that is not of its own colour. Therefore, the Rectangle may go through green platforms and the Circle through yellow ones (see Figure 3). In addition, the platforms may be static of moving in a predefined pattern and collecting diamonds may trigger changes in the level, typically, making platforms disappear. These features present opportunities to create puzzle situations in the game.

III. Challenges for the AI

Several features of Geometry Friends make it a particularly interesting AI testbed, the most important being the cooperative problem solving between the two individual players. For human players cooperation comes naturally as a form of teamwork and communication as way of achieving a common objective. Typically, to succeed in Geometry Friends players need to:

- Determine which collectibles (diamonds) each character can get by itself and which require joint action of both players;
- Determine the order by which the collectibles must be acquired, because certain actions may lead to unrecoverable situations (e.g. if the Rectangle falls it cannot jump back);
- Divide the task of collecting the diamonds by the two players, according to the restrictions determined in the two previous points;
- For each collectible, determine the sequence of controls necessary for the corresponding characters to acquire it, including coordinated actions;
- Execute the controls in the game environment, taking into account the simulated physics;
- Do all these in real-time and as fast as possible.

For all of the above it is implicit that, to perform well, players need to reach some agreement and common commitment for parts of the task.

An AI player will need to deal with all the six points, which present challenges in three different dimensions:

- Dealing with coordination at different levels: from motion control to shared planning;
- Dealing with fine-grained physics-based actuation;
- Dealing with puzzle solving.

Since the two characters have different actuation capabilities, they move in the game world in different ways and, due to the nature of the levels, they often need to join efforts to take advantage of each other's strengths. This collaboration is needed both at planning level, for dividing the task to increase performance, and at actuation level, to coordinate joint actions to reach certain parts of the game world.

Furthermore, although Geometry Friends is viewed and played like most other two dimensional platform games, such as Super Mario Bros., the control of the movement is dynamic and challenging, due to the simulated physics engine (see Section V for details on the players' controls). In particular, there are many situations that require precise timing for simultaneous coordinated actions, as illustrated in Figure 4, for example.

Finally, Geometry Friends, as a puzzle game, features various levels where players must use some strategic reasoning, because there are constraints in the order by which collectibles must be picked. This means that doing a wrong, irreversible action (e.g falling from an platform too soon) can prevent finishing a level with success (see Figure 5 for an example).

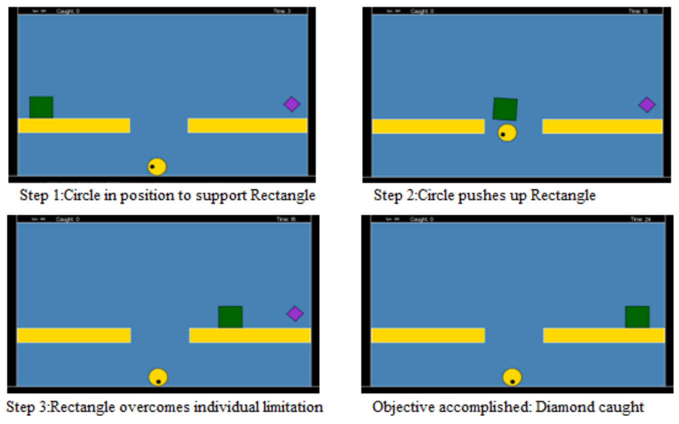
Step 1:Circle in position to support Rectangle
Step 2:Circle pushes up Rectangle
Step 3:Rectangle overcomes individual limitation
Objective accomplished: Diamond caught

Fig. 4. Example of a skill-based cooperative action. In this case the Circle is helping the Rectangle to reach the other side. This requires good timing and mutual action.

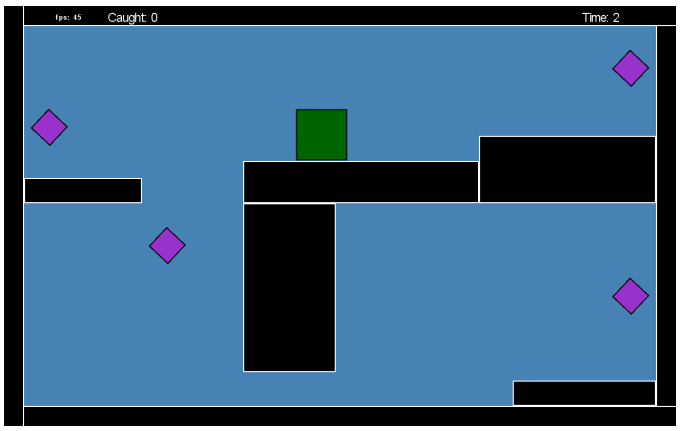

Fig. 5. Example of a level with some puzzle elements. In this level there is an order restriction for the actions of the Rectangle. It needs to go first for the collectibles on the top level (e.g. first to the right and then to the left) and only afterwards to the ones on the lower level.

IV. The Competition

The Geometry Friends Game AI competition includes two main tracks: the **Cooperative Track** and the **Single AI Track** that is divided in two sub-tracks: the **Circle Track** and the **Rectangle Track**. Although the main track is the Cooperative Track, we also include single player tracks for participants that want to tackle the puzzle problem-solving and the characters' control issues before undertaking the more demanding task of cooperation because, to excel in the cooperation task, agents will need good individual control and problem-solving capabilities.

Participants are free to use any approach and algorithm they believe will solve the levels, while baring in mind that the challenge is real-time. For the competition we set a time limit for the resolution of the levels and, at the moment, are using neither moving platforms nor events triggered by collecting diamonds.

The game includes a level editor that participants may use to generate levels with specific situations in which to test their agents.

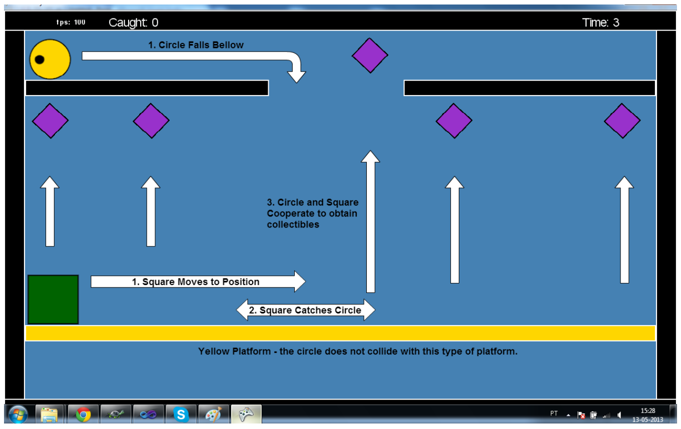
Caught: 0
Time: 3
1. Circle Falls Bellow
3. Circle and Square Cooperate to obtain collectibles
1. Square Moves to Position
2. Square Catches Circle
Yellow Platform - the circle does not collide with this type of platform.

Fig. 6. Cooperative Track level example and potential solution.

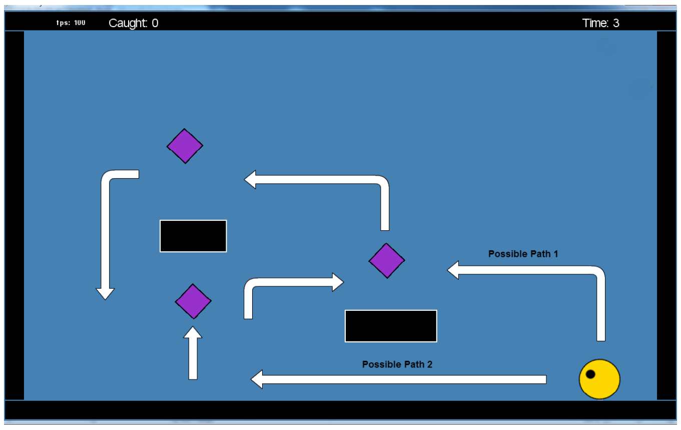

Fig. 7. Circle Track level example and potential solution.

A. *Cooperative Track*

The Cooperative Track focuses on developing agents capable of cooperating with each other in order to solve levels designed for player cooperation. The levels used in this track always require some form of collaboration between both agents. Therefore, participants will have to submit two agents to control each of the two characters (Circle and Rectangle). We discourage the use of direct communication between the two agents, but the framework does not block that at the moment. Figure 6 presents an example of a level in the Cooperative track, where the Circle must use the Rectangle as a jumping platform to solve the level.

B. *Single AI: Circle Track*

In this track participants have to submit an agent to control the Circle character. The Circle character is more challenging in terms of actuation/motor control, since the jump and rolling motions are harder to control and predict than the actions of the Rectangle. Therefore, levels in this track are, primarily, skill and precision challenges. Puzzle elements are still present, but the critical challenge typically requires good precision of the Circle agent actions. Figure 7 depicts an example of a level for this track, displaying multiple paths for solving the level. This level is solved by combining the two actions available to the Circle, rolling and jumping, to jump in diagonal directions.

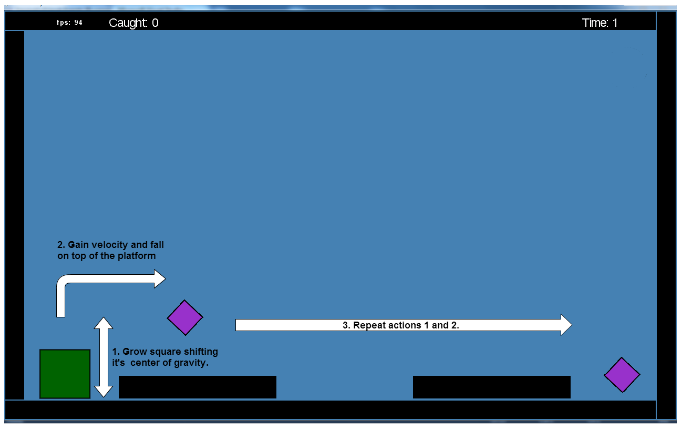
tps: 94
Caught: 0
Time: 1
2. Gain velocity and fall on top of the platform
3. Repeat actions 1 and 2.
1. Grow square shifting it's center of gravity.

Fig. 8. Rectangle Track level example and potential solution.

C. Single AI: Rectangle Track

In the Rectangle track participants have to submit an agent to control the Rectangle character. The motion control of the Rectangle character is easier, since it cannot jump and the corresponding motions are more controlled. For this reason, levels in this track are centred in puzzle challenges, with critical dependencies in terms of the order by which diamonds should be collected. Figure 8 presents an example of level for the Rectangle track, where the character must use gravity and tilt to climb the platforms. Tilt can be achieved by moving left, then morphing up and moving right to make the Rectangle fall over the platform.

D. Ranking

The ranking process ensures an unbiased comparison between all competitors. Submitted agents are evaluated and checked by the organisation committee of the competition. The evaluation of each submission consists of evaluating the performance of the corresponding agent(s) in a set of ten levels, previously defined by the competition organizers. Each track has its own set of levels. These levels consist of a mixture of key challenges that test the agents of the specific track. The set of levels is equal for all the submissions.

Furthermore, five of the levels are disclosed and made available in the competition website at the beginning of the submission process, to help the participants in their development. The other five remain unknown and are made available only after the submission period closes. Keeping half of the levels hidden during the submission period promotes general solutions against solutions that are "over-specialised" to the disclosed levels.

Each level has a time limit and, if the agents solution takes longer than this limit, the level is considered incomplete and the score is computed accordingly. Each submission is evaluated in each level a total of ten times, to reduce the effects of chance due to the real-time execution of the physics engine. This results in a total of 100 runs per submission.

Agents are ranked by the total number of collectibles that they gather in each level as well as by the time they take to complete the level. The final score of a level is, therefore, the average score across the 10 runs of the level. The score of run i is computed as:

$$\text{SCORE}_i = V_{completed} \times \frac{(T_{max} - t)}{T_{max}} + (V_{collect} \times N_{collect}), \quad (1)$$

where, $V_{collect}$ is the score attributed to the agent (or team) for each collected diamond, $N_{collect}$ is the total number of diamonds collected within the time limit, $V_{completed}$ is a bonus score attributed to the agent (or team) if it successfully completes the level (e.g. collects all diamonds) within the time limit, and t is the time it took the agent (or team) to complete the level. T_{max} is the time limit of the level.

The values of $V_{collect}$, $V_{completed}$ and T_{max} are provided to the participants as part of the information associated with each level. In the 2014 Competition, the values of $V_{collect}$ and $V_{completed}$ were the same for all levels, namely 100 and $1,000$ points, respectively. The time limit was different for each level and determined beforehand, based on empirical estimates obtained from the performance of human players playing the game. The resulting time limits ranged from 35 to 120 seconds.

Based on our experience in running the 2014 Competition, we realised that $V_{completed}$ was set too high, favoring agents that solved a couple of simple levels and completely failing the remaining levels over agents that would partially solving most levels even if finishing none (e.g., by collecting more diamonds overall). We still believe that having a bonus for completing a level is useful, as are the time-completion rewards, but the values have been tuned for the 2015 edition of the competition. In addition to this change, we also weight differently the performance on the public and private levels, to further discourage over-specialisation on the public levels. The score of each run of a level is still computed as in (1), but the total score value will be halved for public levels.

V. The GF Agents Framework

Participants have access to a framework to create the AI players/agents, for their submissions. The framework offers C# interfaces and key methods for the agents to use. Example agents are also given in the competition website.

Two distinct player interfaces are defined, one for each character (`IRectangle` and `ICircle`), sharing many features and differing only on the set of actions they accept. The common features are defined in the parent `IAgent` interface that offers three distinct methods: `setup()`, `sensorsUpdated()` and `getAction()`. The `setup()` method is called only once, as the agent starts, and is used for internal configuration procedures; `sensorsUpdated()` and `getAction()` are called on regular intervals with a constant rate. The first notifies the agent that the information on its sensors is ready for the current time-frame and the second gets information regarding the actuation to perform. At setup time, the agent gets the initial state of the level (e.g. position of the elements).

The sensors available give the following information:

- *Platform Information:* platforms' positions and colour, and the level borders;

- *Collectible Information:* diamonds' positions and amount collected;
- *Character Information:* current position (for both characters);
- *Time Information:* time limit and current time;

The actuation is achieved through on/off triggers that control forces acting on the characters. The `getAction()` method gets a set of flags that determine the controls that are on or off. Some triggers may cancel each other (e.g. roll left and roll right). The available controls are:

- *No Action (Both):* cancels all controls, does not do any actuation. The player may keep its movement due to current velocity;
- *Roll Left (Circle):* apply a constant torque to the Circle character making it rolling left, it increases angular speed until it reaches its maximum. The Circle may not start rolling to the left immediately, since it may have a counter rotation;
- *Roll Right (Circle):* similar to Roll Left but applies the torque in the opposite direction;
- *Jump (Circle):* applies a constant force that makes the Circle character jump. The force is only applied when the Circle character is touching the "ground". If active when the Circle is in the air it does nothing;
- *Move Left (Rectangle):* applies a constant force that pushes the Rectangle character to the left. As in the case of the Roll controls of the Circle character, the movement may not start immediately to the left because of the current speed, and the velocity increases until it reaches its limit. Friction has more effect on the Rectangle, because it slides;
- *Move Right (Rectangle):* similar to Move Left, but pushing the Rectangle to the right;
- *Morph Up (Rectangle):* reshapes the Rectangle character, at a constant rate, increasing its height and reducing its width to keep a constant area. While on, the control is performed gradually until the height limit is reached. It can be performed while in the air.
- *Morph Down (Rectangle):* similar to Morph Up but decreases the Rectangle height;

Note that a stop action is not available. In addition, participants should be aware that characters may move without directly activating any movement controls, due to other moving objects in the game world. The constants used in the physics engine, such as, gravity, friction, and the force and torque values used in the characters' controls are provided in the API.

A. *Technical Requirements and Faults*

As stated above, we do not impose any restrictions on the algorithms, approaches or technology used in the agents' implementation. For example, agents may create multiple threads, load and save files (e.g. for learning) and use as much memory and processing power as they want. The only restriction imposed in on the time spent in solving each level: as soon as the time limit is reached, the agents no longer receive the bonus for completing the level. Moreover, if the agent crashes, the run is considered faulted and gets a score of 0 points. Nevertheless, we publish on the website the specifications of the machine where the official tests are run. Essentially, we use a state-of-the-art PC with no special specifications. Participants are disqualified for altering the framework or the game in any way or if the agents do not run at all.

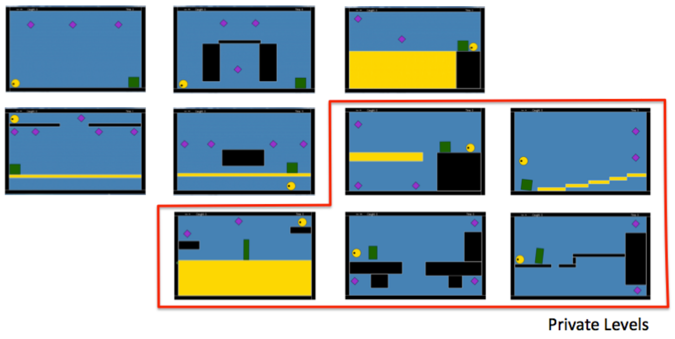

Fig. 9. Levels for the Cooperation Track.

VI. The 2014 Edition

In 2014, the Geometry Friends Game AI Competition was one of the official competitions at the IEEE Conference on Computational Intelligence and Games (CIG), held in Dortmund. It received a total of 6 submissions: 1 for the Cooperative Track, 2 for the Circle Track and 3 for the Rectangle Track. The levels used for each track are presented in Figures 9, 10 and 11.

A. *Agents submitted*

The cooperation track received only one submission:

- **CIBot** (Sejong University): the agents in this team first try to collect as many diamonds as possible on their own. For that they use the same approaches used to solve the single-player tracks (see below). After collecting all diamonds individually, the agents enter cooperation mode. In this mode the Circle agent leads the task. It assumes that the solution is based on the Rectangle serving as a jumping platform for the Circle. The Circle selects the closest diamond to its position as the target to collect and moves there based on its single player approach (see below). The Rectangle, which is also in cooperation mode, follows the Circle, always trying to be below it and, as soon as it gets below a diamond, it stretches to reduce the distance between the Circle and the diamond. The two agents try to move together by "synchronising" their speed. As the diamond is collected, the Circle agent selects a new one and the process is repeated.

The Circle track received two submissions:

- **CIBot** (Sejong University): this agent creates a directed graph in the beginning of the level and uses the Dijkstra's algorithm to find the shortest path through the graph. The graph uses the edges of the platforms as

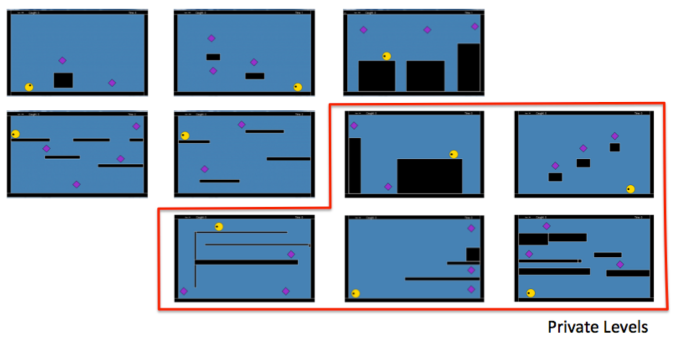

Fig. 10. Levels for the Circle Track.

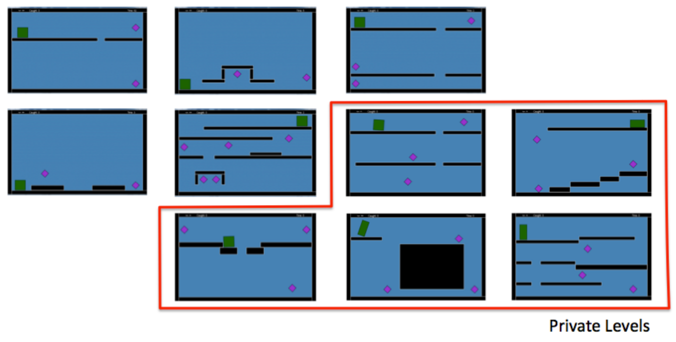

Fig. 11. Levels for the Rectangle Track.

nodes. Connections are created whenever it is possible for the Circle to go from one of those edges to another, e.g. edges of the same platform are always connected. The level "floor" (e.g. the bottom border) is included in the graph. Running Dijkstra's algorithm produces a queue of diamonds ordered by the distance to the initial position of the Circle. The Circle then takes the first as the target and moves to the next once it collects the current target. It repeats this process until the queue is empty (e.g. level solved). To move the character, the agent uses a simple rule based-system that rolls and jumps in a greedy way to move the Circle closer to the target position. Jumping strategies (i.e. the distance and velocity to jump to a platform) are pre-computed in these rules.

- **KUAS-IS Lab** (National Kaohsiung University of Applied Sciences): this agent uses A^* search and Q-learning to solve the levels. A^* search is used to find the shortest path to go through all the diamonds in the level. This is based on a graph that represents the level. The approach is similar to that of CIBot, although it includes the diamonds' positions. The search heuristic uses the distance to the collectible; however, the agent tries to avoid situations that lead to pitfalls by following a greedy approach—for example, if going to the closest diamond renders the level impossible (e.g. in the case the other diamonds become unreachable due to the end position of the character after collecting the diamond). To address this problem, the search heuristic is weighted by the values of a Q-table that reinforces the correct paths and was acquired beforehand by training with Q-Learning.

The Rectangle track received three submissions.

- **CIBot** (Sejong University): this agent uses Monte-Carlo Tree Search (MCTS) to find the best path to collect all diamonds. The agent first analyses the level and defines a directed graph that represents the level. The approach is similar to the CIBot Circle agent, as it uses edges of the platforms as nodes and connects all nodes if it is possible to move from one to the other. The difference is that the traversal check takes into account the shape of the Rectangle. Each connection has information if it allows movement for three possible shapes: fully stretched to the top, squared and fully stretched to the bottom. This allows situations when the movements are only possible if the Rectangle is stretched, e.g. going bellow obstacles. The graph computed is turned into an equivalent tree and the decision is made using the upper confidence bound MCTS with a heuristic that uses the number of diamonds collected and the depth in the tree search.

- **KUAS-IS Lab** (National Kaohsiung University of Applied Sciences): this agent uses the same approach as the KUAS-IS Lab Circle agent. It uses A^* search weighted by a learned Q-table to solve the levels.

- **OPU-SCOM** (Osaka Prefecture University): this agent is composed of two layers. The first layer searches for a global strategy while the second searches for the sequence of actions to complete the strategy. In the first layer the level is converted into a graph by generating cells that cover the level. Cells are generated by tracing lines aligned with all platforms' edges. Each cell becomes a node in the graph and neighbouring cells are connected. Nodes are added for the diamonds and for the initial position of the character. The latter are connected to the node that represents the cell they are in. Dijkstra's algorithm is applied to define possible paths between the character an the diamonds and a particle swarm optimisation algorithm (PSO) searches for the best order between the diamonds, given the possible paths. The agent selects the first diamond on the ordered list returned by the PSO and defines a hierarchical task plan. The plan's lower level represents the elementary actions provided by the game that are executed in the order that the plan specifies. The meta-tasks represent goals, such as catching a diamond or falling over.

B. *Results*

The CIBot was the winner in all three tracks. The results of the Cooperation track are presented in Table I. The results of the Circle track are presented in Tables II and III. The results of the Rectangle track are presented in Tables IV through VI. These tables show for each level the number of runs the agents complete successfully, the average number of diamonds collected (and the total in parenthesis), the average time spent (and the time limit in parenthesis) and the score computed by the Formula (1). Note that levels 1 to 5 are the public levels and levels 6 to 10 are the private ones.

TABLE I. RESULTS OF THE COOPERATIVE TRACK: CIBOT

Level	Runs Completed	Diamonds	Time (Limit) – sec.	Score
1	10	3 (3)	31.66 (90)	948
2	10	3 (3)	32.49 (90)	939
3	10	2 (2)	13.83 (35)	805
4	10	5 (5)	62.03 (110)	936
5	10	4 (4)	74.37 (100)	656
6	0	0 (3)	0 (60)	0
7	0	1 (2)	60 (60)	100
8	0	0 (2)	90 (90)	0
9	0	0 (3)	55 (55)	0
10	0	1 (2)	35 (35)	100
			TOTAL SCORE	4484

TABLE II. RESULTS OF THE CIRCLE TRACK: CIBOT

Level	Runs Completed	Diamonds	Time (Limit) – sec.	Score
1	10	2 (2)	12.67 (20)	567
2	10	3 (3)	19.89 (45)	858
3	10	3 (3)	14.84 (60)	1053
4	0	1.2 (4)	80 (80)	120
5	0	1 (2)	70 (70)	100
6	0	1 (2)	40 (40)	100
7	10	3 (3)	26.19 (60)	864
8	0	0 (3)	40 (40)	0
9	10	3 (3)	50.00 (80)	675
10	0	0 (3)	100 (100)	0
			TOTAL SCORE	4337

TABLE III. RESULTS OF THE CIRCLE TRACK: KUAS-IS LAB

Level	Runs Completed	Diamonds	Time (Limit) – sec.	Score
1	10	2 (2)	5.81 (20)	910
2	0	2 (3)	45 (45)	200
3	0	0 (3)	60 (60)	0
4	0	1 (4)	80 (80)	100
5	0	0 (4)	0 (70)	0
6	0	0 (2)	0 (40)	0
7	0	0 (2)	60 (60)	0
8	0	0 (3)	0 (40)	0
9	0	0 (3)	0 (80)	0
10	0	0 (3)	0 (100)	0
			TOTAL SCORE	1210

TABLE IV. RESULTS OF THE RECTANGLE TRACK: CIBOT

Level	Runs Completed	Diamonds	Time (Limit) – sec.	Score
1	10	2 (2)	12.46 (40)	889
2	10	2 (2)	10.05 (25)	798
3	9	2.9 (3)	32.83 (80)	880
4	10	2 (2)	9.06 (20)	747
5	10	5 (5)	41.64 (90)	1037
6	0	1 (3)	40 (40)	100
7	10	3 (3)	20.93 (50)	881
8	10	3 (3)	21.95 (60)	934
9	0	2 (3)	35 (35)	200
10	0	0 (3)	35 (35)	0
			TOTAL SCORE	6466

TABLE V. RESULTS OF THE RECTANGLE TRACK: KUAS-IS LAB

Level	Runs Completed	Diamonds	Time (Limit) – sec.	Score
1	0	1 (2)	40 (40)	100
2	6	1.6 (2)	20.97 (25)	321
3	0	1 (3)	80 (80)	100
4	9	1.8 (2)	10.53 (20)	653
5	0	2.7 (2)	90 (90)	270
6	0	0.7 (3)	28.00 (40)	70
7	3	2 (3)	37.89 (50)	342
8	6	2.4 (3)	38.98 (60)	590
9	0	0 (3)	0 (35)	0
10	0	0.8 (3)	35 (35)	80
			TOTAL SCORE	2526

TABLE VI. RESULTS OF THE RECTANGLE TRACK: OPU-SCOM

Level	Runs Completed	Diamonds	Time (Limit) – sec.	Score
1	10	2 (2)	12.12 (40)	897
2	10	2 (2)	8.34 (25)	866
3	10	3 (3)	23.17 (80)	1010
4	10	2 (2)	10.79 (20)	661
5	0	1 (5)	90 (90)	100
6	10	3 (3)	19.68 (40)	808
7	0	2 (3)	50.00(50)	200
8	0	1.8 (3)	54.00 (60)	180
9	10	3 (3)	19.14 (35)	753
10	0	0 (3)	35 (35)	0
			TOTAL SCORE	5475

C. Discussion

We can see from the results that the different teams all exhibit some degree of over-specialisation to the public levels. This is particularly prominent in the Cooperation track, where the team performed quite well in the 5 public levels, but poorly in the 5 hidden ones.

The KUAS-IS Lab approach in both tracks (Circle and Rectangle) used a greedy search that often led the character to irreversible situations that rendered the level impossible. The Q-learning strategy to avoid this was not effective. This may be due to insufficient training. The agent might need more training examples to adequately generalise its strategy and effectively avoid the pitfalls.

Interestingly, the OPU-SCOM Rectangle agent got higher scores than the winner (CIBot) in many levels. This happened for levels 1, 2, 3, 6 and 9. But, it solved one less level than the CIBot, pushing it to the second place. It was not able to solve levels 5, 7, 8 and 10. For the ones it did solve, it only got worst performance in level 4. This seems a promising approach. In fact, the two approaches seem complementary. CIBot solved some levels that OPU-SCOM was not able to (levels 7 and 8) and OPU-SCOM solved levels 6 and 9 that CIBot did not. Levels 7 and 8 are more focused on fine motor control rather than level planning, as both include "stairs". Levels 6 and 9, on the other hand, impose restrictions in the order by which diamonds are collected, making these levels more planning-oriented rather than motor control-oriented. This seems to be the main difference between the two agents: CIBot seems more competent at the actuation level and OPU-SCOM seems more competent at the planning level. However, solving the Geometry Friends levels requires both types of competence, reinforcing our conviction that the two approaches (CIBot and OPU-SCOM) are complementary.

Nevertheless, no participant managed to solve all levels, and there are still many opportunities for improvement in all tracks.

VII. Future Extensions

In this section we describe future extensions that are currently being developed for the Geometry Friends Game AI competition. First of all, the game has some features that have yet to be introduced, such as moving platforms. These features are implemented in the game engine but left out of the competition levels, as we are currently waiting to get better results in the tracks as they are before adding more complexity to the base problem.

The competition may include a procedural-generated level track as well, which we believe would be an interesting scenario for the development of PCG (Procedural Content Generated) algorithms. The creation of puzzle game scenarios is not something completely new in the PCG community, although we believe that the game Geometry Friends will prove an interesting test-bed for these algorithms. The main novelty is the cooperative nature of Geometry Friends, which adds some interesting challenges, as levels should to be fun for both players. Additionally, making sure that the levels are solvable may actually be a hard problem. We already have a clear specification for the levels, which would facilitate the development of this track. Nevertheless, we would still require an adequate framework to evaluate the levels generated. One possibility would be to follow the current evaluation practices in the level generation track of the Mario AI competition [11]. The ranking process would consist of interleaving artificially-generated levels with human-created levels and allow human players to play all. At the end of the play session, each player would then be asked to rank the levels accord to their preference and identify which levels were algorithmically created and which levels were not.

Another extension that has been considered is an Agent Believability track. This track would consist of creating agents that would act in a human-believable way. The ranking process could consist of a Turing-like test where users would view various human and agent game-play sessions and identify the artificial agent from the human player.

Finally, the last idea currently being developed, which is in fact the main motivation for the development of the whole agent framework, is the Human & AI Cooperation track. This will consist of developing a single AI agent (Circle or Rectangle) that can play cooperatively with a human player. Besides the challenges presented in the previous sections, this track would require agents to effectively communicate with human players, predict their movements and interact with their characters in an entertaining way. The ranking of this track would require tracking the performance of both players (such as the number of levels solved and how long did it take to solve them) and randomly have users play with other users or agents (without their knowledge) and ask them how their playing experience was. This will include similar concerns as the believability track.

VIII. Conclusion

We are convinced that the Geometry Friends Game AI Competition will be a successful and challenging test-bed for single and cooperative AI agents. Although a simple-looking game, Geometry Friends presents a variety of interesting and original challenges that can increase in difficulty through ingenious level design and the various combinations of its simple mechanics. Current approaches are still far from solving all the competition levels, which indicates that there are still many open challenges to tackle, in particular, in the cooperative setting. There are also many opportunities to extend the competition. One that we plan in the near future is the interaction with human players that we believe will foster research in human-agent interaction.

Acknowledgment

This work was supported by FCT – Fundação para a Ciência e a Tecnologia (INESC-ID multi annual funding) under project UID/CEC/50021/2013. We would also like to thank to all participants of the competition for their submissions.

References

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *Proceedings of the first international conference on Autonomous agents*. ACM, 1997, pp. 340–347.

[2] R. Arunachalam and N. M. Sadeh, "The supply chain trading agent competition," *Electronic Commerce Research and Applications*, vol. 4, no. 1, pp. 66–84, 2005.

[3] G. Seetharaman, A. Lakhotia, and E. P. Blasch, "Unmanned vehicles come of age: The darpa grand challenge," *Computer*, vol. 39, no. 12, pp. 26–29, 2006.

[4] D. Billings, D. Papp, J. Schaeffer, and D. Szafron, "Poker as a testbed for ai research," *Advances in Artificial Intelligence*, pp. 228–238, 1998.

[5] M. Genesereth, N. Love, and B. Pell, "General game playing: Overview of the aaai competition," *AI Magazine*, vol. 26, no. 2, pp. 62–72, 2005.

[6] J. Togelius, S. Karakovskiy, and R. Baumgarten, "The 2009 mario ai competition," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2010.

[7] P. Hingston, "A turing test for computer game bots," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 3, pp. 169–186, 2009.

[8] D. Loiacono, P. L. Lanzi, J. Togelius, E. Onieva, D. A. Pelta, M. V. Butz, T. D. Lonneker, L. Cardamone, D. Perez, Y. Sáez *et al.*, "The 2009 simulated car racing championship," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 2, no. 2, pp. 131–147, 2010.

[9] P. Rohlfshagen and S. Lucas, "Ms pac-man versus ghost team cec 2011 competition," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2011, pp. 70–77.

[10] J. B. Rocha, S. Mascarenhas, and R. Prada, "Game mechanics for cooperative games," *ZON Digital Games 2008*, pp. 72–80, 2008.

[11] N. Shaker, J. Togelius, G. N. Yannakakis, B. Weber, T. Shimizu, T. Hashiyama, N. Sorenson, P. Pasquier, P. Mawhorter, G. Takahashi *et al.*, "The 2010 mario ai championship: Level generation track," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 4, pp. 332–347, 2011.