# A Platform for the Geometry Friends Cooperative Game AI Competition

**RICARDO COSTA,[1] RUI PRADA,[2] FRANCISCO S. MELO[3]**

ABSTRACT

This paper describes the development of a new online competition platform to support the future of the Geometry Friends Cooperative Game AI competition. Geometry Friends is a cooperative 2D physics-based platformer that presents several challenges for the AI community, in particular, to integrate motion and task planning. This paper presents the work developed to improve the platform used to run the competition. The work was based on best practices of similar platforms. The new platform was developed mostly from scratch, consisting of a new website and a background program in charge of fully handling the received submissions, and more. Virtualization was used to create a secure, fair and reusable evaluation environment. Different features of the new platform were tested by several people across three different test scenarios. The new website was concluded to have an above average usability, while the submission handling program worked as expected throughout the testing phase.

KEYWORDS

Video Game Competitions;
Artificial Intelligence;
Cooperative Games;
Automation.

[1] INESC-ID and Instituto Superior Técnico, Univ. de Lisboa, Portugal — ricardo.gomes.costa@tecnico.ulisboa.pt

[2] INESC-ID and Instituto Superior Técnico, Univ. de Lisboa, Portugal — rui.prada@tecnico.ulisboa.pt

[3] INESC-ID and Instituto Superior Técnico, Univ. de Lisboa, Portugal — fmelo@inesc-id.pt

## 1. INTRODUCTION

Videogames make one of the best testbeds for artificial intelligence. They provide inexpensive virtual simulations where algorithms can be tested repeatedly. They can also be viewed as good platforms for iterative AI testing models, with the possibility for gradual improvements over time. Geometry Friends (Rocha et al. 2008) is one of such games. Initially designed to be a 2D platform puzzle game oriented towards cooperation, it was adapted to enable the implementation of artificial agents. A Geometry Friends AI competition (Prada et al. 2015) exists and has been receiving submissions annually since 2013. Competitions allow different parties to present and compare their solutions. Several factors make a competition more attractive, such as: the game quality; unique challenges; and good competition platforms. A competition platform is a system composed by several components such as the interfaces which the participants and even competition organizers interact with, participant registry, solution submission and consequent processing, community interaction, etc.

The competition platform of the Geometry Friends Game AI competition (GFGAI) is the main focus of this work. Improving a competition platform not only facilitates participation, but also cuts costs of running and maintaining a competition. As it follows, by improving the Geometry Friends Game AI Competition platform, the number of submissions per competition edition could increase, as well as the amount of editions that could be held in the future. This is relevant because Geometry Friends is a game that poses a rather unique set of challenges, which are not considered to be completely solved. This means that there is still room for improvement, where new algorithms and ideas can be explored throughout the future history of the GFGAI competition.

## 2. GEOMETRY FRIENDS

Geometry Friends is a 2D puzzle platformer where up to two playing characters — a yellow circle and/or a green rectangle — can move with their unique set of actions to capture purple diamonds. Figure 1 shows a sample level which not only includes the characters and collectibles mentioned above, but also the different platforms available: normal (black) impenetrable platforms; and coloured platforms. A player object will collide with a wall whose colour does not match its own.

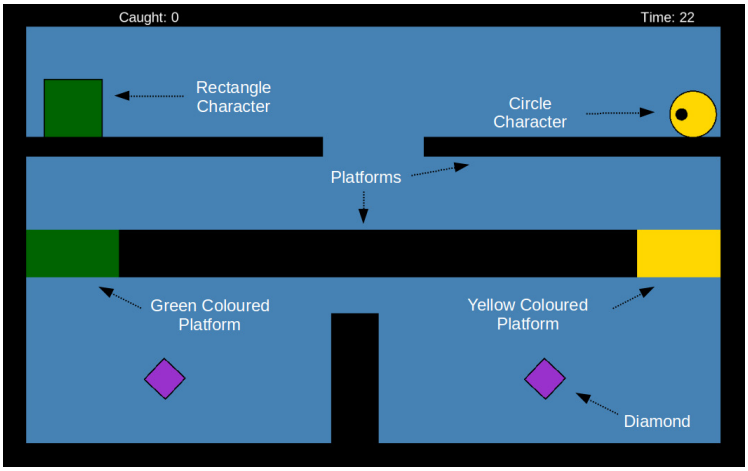


Fig. 1
A Geometry Friends level featuring all elements of the game: the circle and rectangle players, all three types of platforms and the purple diamonds.

One of the most interesting aspects of this game is the cooperation factor in levels, or maps containing both the player characters. It is often required that both the characters combine their efforts and unique abilities to complete the levels.

## 2.1 GEOMETRY FRIENDS COMPETITION

The Geometry Friends Game AI competition allows participants to implement agents which control one of the characters, and to compete against other participants in the same levels, or maps. Agent development can be done using a C# API, as the original game is also fully implemented in C#. Both the framework and sample agents can be found at the competition website.[4] The competition has been successfully held a few times at IEEE Conference on Computation Intelligence and Games.

Participants are asked to zip their solution containing the source code developed and email it to the competition organizers, along with entry details, such as: a team name; intended category (circle only, rectangle only or cooperative); and a small technical report describing the solution. A command line interface and a Geometry Friends Batch Simulator is available to help participants test their solutions and organizers to evaluate submitted solutions.

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

A competition has public and private levels, to discourage overfitting solutions. The score of each entry the sum of the scores it gets in all levels (usually the competition presents 10 different levels). Each agent is run several times in each level. The final score of the level is the average score of the runs. The score of each run depends on the time limit, the time the agent took to solve the level and the number of collectibles it got. This process takes long and, at the time of the beginning of this work, required manual labour. In fact, many of the competition management processes require manual labour, which can scale up with the number of participants. These include the initial setup of the competition, receiving of submissions, handling them (i.e. compiling and evaluating) and updating the results on the competition website.

## 3. GAME AI COMPETITIONS

Artificial Intelligence competitions are not new. They exist as a way to promote advances in the field by encouraging the development of AI based solutions and sharing with members of the community. Popular examples lie in card games like Poker (Billings et al. 1998), real-time strategy (RTS) games such as StarCraft (Buro et al. 2012) and turn based games, such as Pokemon (Lee et al. 2017). Competitions are also common in real-word applications, such as the Supply Chain Trading Agent Competition (Arunachalam 2005), transportation challenges, such as the DARPA Grand Challenge (Seetharaman 2006) and human conversational intelligence, such as the Loebner Prize (Powers 1998).

### 3.1 GOOD PRACTICES

Good practices for running AI competitions are explored and suggested by Togelius (2014). The author attempts to explain the reasons why some competitions fails and then suggests guidelines for the success of AI competitions. As for why AI competitions fail, Togelius defends that the lack of continuity, stagnation and irrelevance are the main reasons. That is, competitions fail when they do not evolve and keep their challenges relevant to the field, and when low amount of effort is put into them, preventing continuity to the competition. To run a successful AI competition, Togelius suggests that a competition should:

- be fully transparent in terms of rules and evaluation methods;
- be accessible on a wide variety of platforms and programming languages;
- be repeated to enable improvements over time;
- have a discussion group to encourage community interaction;
- have software that can run locally to test solutions more efficiently;

- have a game that can be sped up, useful to train learning algorithms;
- be easy on beginners, e.g. including sample agents and simple instructions;
- open-source everything, including solutions, to enable sharing and prevent cheating.

## 3.1 GAME COMPETITIONS PLATFORMS

During this work, several artificial intelligence competition platforms were studied. One of them, the Mario AI championship, is based on the Mario AI Benchmark, which is itself based on the Infinite Mario Bros game developed and made open-source by Markus Persson (Karakovskiy et al. 2012; Togelius et al. 2013). Active between 2009 and 2012, the competition allowed agents to be developed in multiple languages, provided good documentation and community interaction via a Google site and a Google group. Submissions were sent via email. Another platform is OpenAI's Gym, which is not necessarily a competition, but allows the submission, scoring and sharing of solutions based on reinforcement learning (Brockman 2016). A more recent platform called Universe was developed on top of Gym to enable agents to mimic the actions of a computer user, i.e. by simulating keyboard and mouse events. An interesting feature of the OpenAI Gym's framework was the ability to automatically upload a solution to the platform using code.

The General Video Game AI (GVGAI) Competition, which like the OpenAI Gym promotes game agnostic solutions (Perez-Liebana et al. 2016), is a modern competition platform that follows many of the guidelines suggested in Section 3.1. It allows for submissions to be uploaded through the website. Its framework is cross-platform and agents may be developed in multiple languages (Java and Python). In a paper, the authors give an insight into the back-end processes of the platform (Perez et al. 2015) which we summarize and illustrate in Figure 2. An automated process such as this ensures high repeatability and low organizational costs.

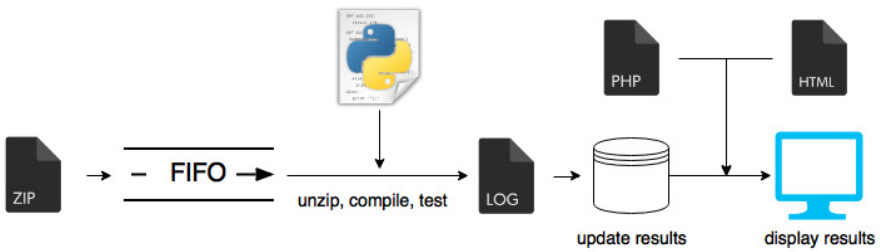


Fig. 2
A simplified diagram which describes the back-end processes of the gvgai competition (1-player tracks only)

As for the GVGAI results tables (see figure 3), they manage to act like a hub for lots of information regarding the submissions, such as score, participant profile and source-code download.

Test Set 2-Player Games CEC 2017

| Rank | Username | User ID | Country | Points | Avg. Glicko Score | Games Played | Controller |
|---|---|---|---|---|---|---|---|
| 1 | ToVo2 | 474 | Slovenia | 161 | 1183.41 | 5070 | Download |
| 2 | ehauckdo | 542 | Brazil | 111 | 1133.4 | 6338 | Download |
| 3 | not2048 | 671 | United Kingdom | 111 | 1071.82 | 5170 | Download |
| 4 | essex_acwebb | 530 | United Kingdom | 110 | 1120.63 | 5070 | Download |
| 5 | Number27 | 441 | Germany | 84 | 1124.69 | 5170 | Download |
| 6 | sampleRS Sample | 636 | United Kingdom | 78 | 1108.92 | 5349 | Download |

Fig. 3
Part of a table showing the 2-player track overall results of CEC 2017, as indicated by the label above the table. The first row explains what each column represents. Both the usernames and the "Download" are hyperlinks. From http://www.gvgai.net/gvg_rankings_conf_2p.php?rg=2006

Lastly, the Ms. Pac-Man Vs Ghost Team is a competition based on the popular arcade game Ms. Pac-Man (Williams 2016). Its website features a thorough step-by-step image-based guide and also allows for submission uploads through the website. One of the features that stands out, however, is the existence of a controller packaging script, a Bash script that compresses the participant's solution into a single, submission ready file.

## 4. IMPROVING THE GEOMETRY FRIENDS COMPETITION PLATFORM

The Geometry Friends competition meets some of the good practices, such as, supporting the ability to test the game locally. However, it also had several weaknesses, in particular, the fact that a lot of (human) labour was needed in the organization processes, which was proportional to the amount of submissions per competition edition, or instance. This often led to delays in the availability of competition related information, e.g. past submissions and technical reports, and result tables.

There are three main groups of people that interact with the Geometry Friends competition: the organizers; the participants; and the general public. When considering the main weaknesses previously presented, it is possible to present requirements that the new platform should meet for each of the three groups.

Most of this work focuses on improving the processes related to the organization of the GFGAI competition. With the new platform, an organizer should be able to: create new competitions and configure parameters such as the name, start and end dates, evaluation formula parameters for each level, etc.; easily run predefined submissions to each competition to serve as baselines for the partic-

ipants; have all the submission evaluation processes automated for them, including the reception of submission files, extraction, compilation, execution, obtaining results and making them publicly available, etc.; and be able to update the GF game version in use when evaluating submissions, in case the game itself evolves in the future.

On the other side of things, a participant should be able to: understand if and what competition editions and categories are open for submissions; access competition details and parameters such as levels, time limits and bonuses for each level, etc.; have rapid access to the GFGAI competition framework and quickly create a simple agent; upload a submission for a desired competition instance automatically; understand what happened to his submissions (e.g. possible errors) and what their results were; submit multiple times to a single competition (given that only the latest submission counts); and create and submit more complex submissions which may include C# dependencies and even platform specific dependencies.

Lastly, any person should be able to: have access to all current and past competition details (e.g. levels used, formula parameters) and results/scoreboards; and have access to all submission files and technical reports submitted throughout the lifetime of the new platform (except for Competitions which have not finished yet, to prevent cheating).

## 4.1 OVERVIEW OF THE NEW PLATFORM

In an attempt to meet the requirements presented above, a new platform was developed. It was highly inspired by the GVGAI competition. Although, with very specific differences resulting from the unique characteristics of the GFGAI competition.

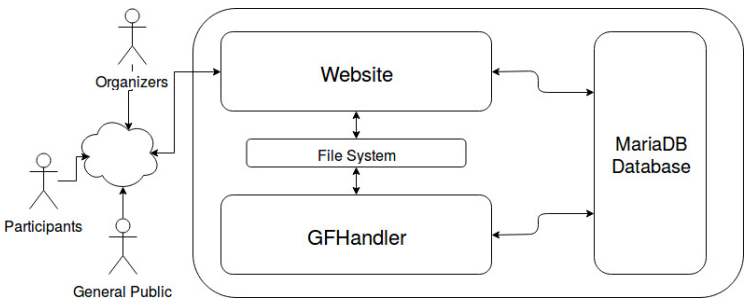


Fig. 4
Conceptualization of the two main components of the new platform.

The new GFGAI competition platform was developed as a web application following a traditional LAMP model. The platform's design was split into two separate and almost independent components,

as shown in Figure 4: the Website the component, which is meant to be interacted with, both by competition participants and administrators; and the GFHandler the background process, which is in charge of the automation of the competition handling processes. They share a MariaDB database, which stores most of the persistent competition related data, such as user information, submission details, existing competitions, etc. The two components also share files in the file system, such as the submission files.

## 5. THE NEW WEBSITE

The new website is one of the two main components introduced in the previous section. It is the only component which interacts directly with its users: organizers, participants and the general public.

### 5.1 MANAGEMENT INTERFACES

The management interfaces are meant to be accessed and used by the organizers alone. Because of this, a special type of user account exists, called an administrator account, which enables the use of these interfaces.

Organizers may create competitions and then configure them (Fig. 5) by editing, for instance, the number of simulations executed per level, the maximum submission size and level specific parameters (visibility and formula related parameters).

Organizers are able to run pre-sets, which can be thought of special "participants" controlled by the organizers, which can "submit" a predetermined submission on command. This is useful to have constant baselines across the different competitions.

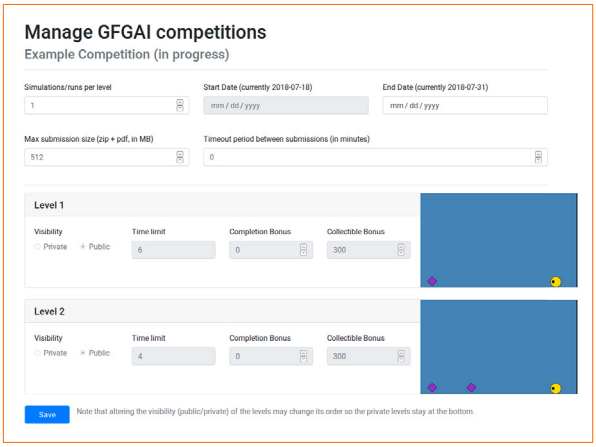


Fig. 5
Management interface to edit competition parameters.

### 5.2 PARTICIPATING

A participant may pick a competition from a list of competitions and download a competition package containing a copy of the game, a sample agent, the XML world file describing the levels of the competition (only the public levels in case the competition in question is still ongoing) and two packaging scripts to facilitate the creation of the files to be submitted. After submitting an entry to the competition, participants can track their results in real time on their profile page, which shows both the score and state of the user's submissions. This state indicates feedback from the GFHandler component, such as if the submission is still being processed, or if it had errors (the system may even make error logs downloadable), etc.

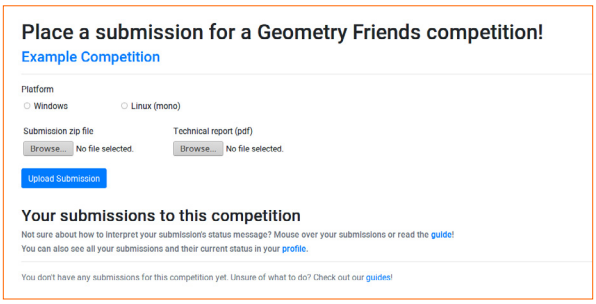


Fig. 6
Submission form to upload a ZIP solution.

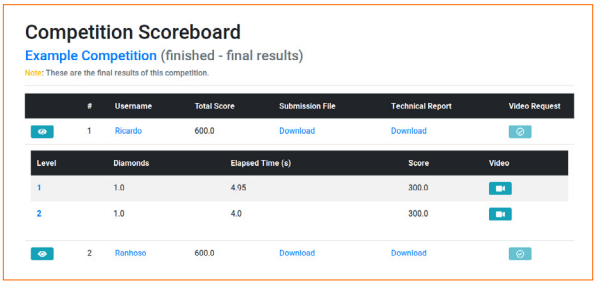


Fig. 7
Scoreboard with two entries. The first entry is expanded to show the results more details.

Finally, users may check the current scoreboard (Figure 7) of the competition they submitted to in order to compare their results against other participants, as well as see their level specific scores and request videos of the execution of their agents.

## 6. GFHANDLER: THE SUBMISSION HANDLING PROGRAM

The most complex component of the developed platform is a background program, which we called GFHandler, since it is meant to handle Geometry Friends submissions. This ended up being a very custom program tailored for the quirks of the Geometry Friends game and framework.

## 6.1 THE MAIN LOOP OF THE GFHANDLER PROGRAM

The main loop regularly polls the database for new (unprocessed) submission entries. These submissions are the ones that are uploaded using the new website, using the forms. Each submission entry in the database possesses all information required by the GFHandler, one of the most important being the path where the uploaded submission was stored to. If there is a new submission to handle, the GFHandler will start a submission environment in a virtual machine. This environment is populated with the necessary files, such as the submission (ZIP file), a fresh copy of the game and the levels (XML file) of the competition, which the submission was placed to. Once everything is in place, a series of events occur automatically, including extraction of the ZIP file, compilation of the source code, simulation of the compiled agent on the several levels (usually multiple times per level, a parameter which can be configured by an organizer using the new website) and extraction of the results from a game generated Results.csv file.

## 6.2 VIRTUAL MACHINES AS VERSATILE, SECURE AND FAIR SIMULATION ENVIRONMENTS

Submissions are handled in environments that encapsulate the submission handling processes in order to: support standardization of the mentioned processes, which may enable all submissions to be handled in the same way and thus simplifying the system; promote fairness, if the environment is guaranteed to be equal for every submission; support modularity, in the sense that the environments may change (e.g. by adapting to future changes of the Geometry Friends framework) without the need to modify the main program or loop of the GFHandler; grant isolation, whose magnitude may vary depending on the environment chosen, but grants a greater security for the host system in case a malicious submission is sent.

Virtualization[5] was chosen as a way to accomplish the desired outcome. Because with virtual machines, we can: run entirely different operating systems on the same machine (e.g. Windows on Linux) and thus avoid the problem of platform specific dependencies; have isolated environments with no access to the host system where even malicious submissions can be executed safely; save a clean state (snapshot) of a virtual machine and then restore

[5] The solution developed uses VirtualBox https://www.virtualbox.org/

a machine to that snapshot every time we handle a submission to ensure a constant and fair environment for each submission.

## 6.3 HANDLING THE SUBMISSIONS USING VM ENVIRONMENTS

The GFHandler program runs commands remotely (via SSH) in either Bash or PowerShell, depending on the environment required (Linux or Windows). A status attribute of the submissions is stored in the DB to allow the GFHandler to see which submissions have already been processed. After detecting the existence of an unprocessed submission, GFHandler decides what levels the submission needs to be simulated on by looking at whether the respective competition has ended or not. Only when a competition ends will the submissions be evaluated on that competition's private levels.

Another flag in the database indicates whether the submission should be run using Windows or Linux. The respective VM is restored to a clean snapshot and is booted. After a while, the GFHandler connects to the VMs via SSH and begins a series of steps:

1. sends the submission ZIP to the VM via SCP;
2. extracts the ZIP (with "unzip" in Ubuntu or "Expand-Archive" in Windows PowerShell);
3. removes unwanted files from the user's submissions (e.g. the game files, in case the participant also sent them in their submission) to prevent cheating with altered files;
4. sends a fresh copy of the game and an XML file containing only the levels that the submission is supposed to be evaluated on;
5. builds the solution using MSBuild,[6] which should generate a DLL file containing the submitted agent;
6. simulates the agent on the levels, R runs per level, computing the score using equation in Section 2.2 by parsing the results from a game generated CSV file;
7. powers off the VM.

On step 6, the score on each level is computed and updated on the database before the agent is simulated on the next level, making it possible to track the progress in real time.

[6] https://msdn.microsoft.com/en-us/library/dd393574.aspx (accessed July 26, 2018)

#### ERROR HANDLING

The GFHandler program is expecting errors at any stage of the handling process. In steps, such as, extraction or compilation, submissions may fail. If so, they are assigned a status message accordingly, like “extraction error” or “compilation error”. Contrary to errors in other stages, the GFHandler does not stop the entire progress after encountering runtime errors during game simulation. This is to give a chance to agents who just crash occasionally to still get some score for the levels where they performed well. For both compilation and runtime errors, the GFHandler program stores the output of the executed commands in a single log file which can later be displayed to the participant through the website. This log file may contain, for example, the output of a MSBuild’s compilation, or even runtime error traces for each individual level the submission failed on.

## 7. EVALUATION

The platform was evaluated in terms of the general usability using a SUS[7] questionnaire. Three test scenarios were designed, one testing the organizational side of things, and the other two the participants side. The competition management (CM) test had testers create and edit a competition. It also had them create and run preset submissions and track their results. A thorough user (TU) test had testers act as participants by creating and logging in with a new account, downloading a competition package (game + sample agent + competition specific levels), creating a simple agent in VisualStudio or MonoDevelop, submitting it to a certain competition and tracking the results. A simple participant (SU) test made available two ready to upload submissions and had the testers create and login with a new account, submit two submissions and track their progress. Testers were also asked to explain what had happened to their first submission after submitting the second.

Each person asked to evaluate the system could do only two of the scenarios, since both TU and SU scenarios had overlapping tasks. As such, each person could test the competition management scenario and one of the competition participant’s scenario. People were invited to perform the tests using their own computers, as a way to also test the platforms across different systems and browsers.

[7] https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

## 7.2 RESULTS

A total of 18 people participated in the tests, from July 20 to September 14. The number of people on each test can be found on Table 1, along with the SUS scores.

| SCENARIO | # TESTS | AVERAGE SUS | Ø |
|---|---|---|---|
| CM | 18 | 76.6(6) | 17.1499 |
| TU | 9 | 78.3(3) | 16.0078 |
| SU | 9 | 84.4(4) | 10.8813 |
| Total | 36 | 79.027(7) | 15.4605 |

Table 1
SUS results for all three test scenarios

All but one of the testers were students between 18 and 25 years old, and 2 were female. Over half of the people tested studied IT related fields. One professor of a relevant field of study, AI for games, also participated.

According to results of 500 SUS evaluations, the average SUS score is 68 and achieving over 80.3 is considered a very good score. This means that an above average score was obtained in all our scenarios, and the SU scenario obtained a very good usability score. This is not surprising as it was designed to be the simplest scenario. However, because the main difference between the TU test and the SU test is the fact that the TU test required interaction with the GF framework, the difference in scores may indicate that the website is more usable than the framework.

As for the tasks of each scenario, everyone was able to complete them, and the majority was able to give the expected answers, meaning they were able to create a simple agent successfully (TU) and understand what happened to their submissions or pre-sets (on all scenarios). On the CM scenario, everyone created and configured a competition correctly as expected. However, a couple of people (1 on the SU and 1 on the CM scenarios) did not give expected answers when reporting on the status of their submissions or pre-sets, mostly because they did not realize that the system was evaluating their submissions in real time.

The professor, who had experience using platforms such as Mooshak, complimented the new platform, more specifically the modern design of the website.

## 8. CONCLUSION

A new and functional platform for the Geometry Friends Game AI competition was developed successfully during this work. The usability of the website was considered above average using a System Usability Scale (SUS). No improvements were required on the platform's second main component, the GFHandler, as it functioned exactly as intended throughout all the tests. All the requirements specified in Section 4.2 were successfully implemented, and most of them tested repeatedly on the aforementioned tests with people. Thus, we can conclude that the main goal was accomplished, since a new platform was indeed designed, implemented and deployed successfully.

Moreover, considerable care was put into helpful documentation not only for the participants and competition organizers (extensive guides on the website), but also to whomever will be in charge of maintaining or continuing the development of the platform itself. The implemented solution takes a different approach from the current state of the art, mostly by using virtualization during submission evaluation processes to guarantee security against malicious foreign code, fairness and more freedom for the participants to develop platform specific code, if they so desire. While this provides a small overhead in submission evaluation times mostly due to booting up the virtual machines, this is still small compared to the rest of the necessary evaluation steps.

## ACKNOWLEDGEMENTS

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT-UID/CEC/500 21/2013) and through the project RAGE (Ref. H2020-ICT-2014-1/644187).

## REFERENCES

**J. B. Rocha, S. Mascarenhas, R. Prada**: *Game mechanics for cooperative games*. ZON Digital Games 2008, pp. 72-80, Porto. 2008.

**R. Prada, P. Lopes, J. Catarino, J. Quitério, F. S. Melo.** *The Geometry Friends Game AI Competition.* Proceedings of CIG'2015 – IEEE Conference on Computational Intelligence and Games, pp. 431-438, Tainan, Taiwan. August 2015. IEEE.

**D. Billings, D. Papp, J. Schaeffer, and D. Szafron**. *Poker as a testbed for machine intelligence research.* Advances in Artificial Intelligence, pp. 228–238. 1998

**M. Buro, D. Churchill.** *Real-time strategy game competitions.* AI Magazine, vol. 33, no. 3, pp. 106–108. 2012

**S. Lee, J. Togelius.** *Showdown AI Competition.* Proceedings of the IEEE Conference on Computational Intelligence in Games, pp. 191-198. 2017

**R. Arunachalam, N. M. Sadeh.** *The supply chain trading agent competition.* Elec-

tronic Commerce Research and Applications, vol. 4, no. 1, pp. 66–84. 2005

**G. Seetharaman, A. Lakhotia, E.P. Blasch.** *Unmanned Vehicles Come of Age: The DARPA Grand Challenge.* Computer 39, 26–29. 2006

**D. Powers.** *The Total Turing Test and the Loebner Prize.* Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP3/CoNLL '98, Association for Computational Linguistics, pp. 279-280. 1998

**J. Togelius.** *How to run a successful game-based AI competition*. IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, pp. 95-100. 2014

**S. Karakovskiy, J. Togelius.** *The Mario AI Benchmark and Competitions.* IEEE Transactions on Computational Intelligence and AI in Games, 4(1):55–67. 2012.

**J. Togelius, N. Shaker, S. Karakovskiy, G. N. Yannakakis.** *The Mario AI Championship 2009-2012*. AI Magazine, vol. 34, no. 3, pp. 89–92. 2013.

**G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba.** *OpenAI Gym*. arXiv preprint arXiv:1606.01540. 2016

**D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, S. Lucas.** *General Video Game AI: Competition, Challenges and Opportunities.* Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 2016

**D. Perez, S. Samothrakis, J. Togelius, T. Schaul, S. Lucas, A. Couetoux, J. Lee, C. Lim, T. Thompson.** *The 2014 General Game Playing Competition*. IEEE Transactions on Computational Intelligence and AI in Games, DOI: 10.1109/TCIAIG.2015.2402393. 2015

**P.R. Williams, D. Perez-Liebana, S.M. Lucas.** *Ms Pac-Man Versus Ghost Team CIG 2016 Competition*. Proceedings of the IEEE Conference on Computational Intelligence and Games. 2016