

Implementing a remote submission system for the Geometry Friends AI competition

Pedro Costa, Hugo Fernandes, Edgar Santos

Activities Report

Abstract—This group worked to develop a robust remote submission system for AI competitions using the Geometry Friends framework. To this end, group elements worked in areas such as front-end and back-end development for submissions, and code review and organization of the framework code repositories, as well as bug fixing. Existing solutions were analyzed by the group members, and new solutions were proposed to the activity promoter. The final submission system was partially implemented, paving the way for future endeavours in this project. The new system allows for flexible submission of agents by the competitors. Further, corrections to the framework now allow for better comparison of agents for grading.

Index Terms—(Geometry Friends, Artificial Intelligence, AI, Competition, Submission System, L^AT_EX).

The document is highly focused on the description of the system rather than the ACTIVITY!

1 INTRODUCTION

GEOMETRY Friends [1] [2] is a cooperative puzzle game that was developed by GAIPS INESC-ID laboratory using Microsoft's XNA framework. The game has a website made and supported in WordPress where anyone can see the game description, learn how to play, download the source or the binaries, and see results of artificial intelligence agents competitions. The goal of the game is to solve

needs to grab all the diamonds that are spread in the level, but to reach the diamonds the players must cooperate to overcome several obstacles in a clever way. In our work, we fo-



Figure 1. Geometry Friends Logo.

different levels using two characters, a circle and a rectangle. To solve one level the players

- Pedro Costa, nr. 49729,
E-mail: pedro.a.r.costa@tecnico.ulisboa.pt,
- Hugo Fernandes, nr. 73665,
E-mail: hugo.apfernandes@gmail.com,
- Edgar Santos, nr. 64753,
E-mail: edfil221@gmail.com,
Instituto Superior Técnico, Universidade de Lisboa.

Manuscript received June 5, 2015.

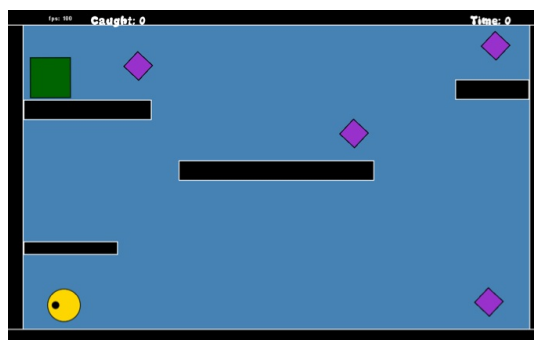


Figure 2. Geometry Friends in-game screenshot.

figures MUST be referenced in Text!

cused mainly in the artificial intelligence competitions, in which any person can design their own solution and create autonomous agents to control the circle and the rectangle to try to conclude the game successfully. The goal of our work was to improve the submission system existing in the game's website and back-end to allow an automatic mechanism to submit and test competitors solutions.

(1.0) Excellent	ACTIVITY						DOCUMENT						
(0.8) Very Good	Object × 2	Opt × 1	Exec × 4	Summ × .5	Concl × .5	SCORE	Struct × .25	Ortog × .25	Exec × 4	Form × .25	Titles × .5	File × .5	SCORE
(0.6) Good	1.0	0.8	0.7	1.0	0.8		1.0	0.8	0.8	1.0	1.0	1.0	
(0.4) Fair													
(0.2) Weak													

2 BACKGROUND

The previous architecture required someone to manually gather the competitors submitted code via email and test it, there was the need to waste time and effort to treat the submitted code manually by someone responsible from the game. There was also a lack of feedback to the participants submissions, in which the code would only be evaluated in the competition's deadline end, due to the work needed to test manually each submission of the players frequently. So the competitors only could know the results in the end of the competition, that means that sometimes even minor errors, for example in compilation, could invalidate a whole work of a participant.

3 PROCEDURE

To design our solution, team members focused in different areas: in the back-end area there was a need to review the correctness of the game and to try to improve its current performance and tests. In the front-end area, there was a need to fully remake several pages of the website, to allow users to use an uploading mechanism and submit feedback. We also made a system that was responsible for the automation of the compiling and execution of the game, making the bridge between the Front-End and the Back-End.

3.1 Developing the Front-End

The first step in front-end development was to detect the necessary changes to be made in terms of website pages to support the new submission system. The existing website, in terms of submission, only had a page explaining the competitors how it was supposed to submit their intelligent agents, that was by sending an email with the source code of their agents and a .pdf file explaining their solution. The idea behind the new submission system was that a competitor uploaded their source code and documentation directly in the website specifying the competition that they are applying to. Then, the uploaded files would be moved to the competition folder in the server together with the competitor information. For

the desired submission system, we needed to provide a way to upload the source code and the documentation directly in the website. To do this, we created a new version of the submission page, which used two WordPress plugins, Wordpress File Upload and Insert PHP. The new page now have two upload buttons, one for source code and the other for documentation. The submitted files are uploaded to the website's server. We also used Insert PHP plug-in to allow a connection to the WordPress database, so that the user can select the competition that they want to apply to. To do this, we set a PHP database connection providing the user name and password and then a query is made to a specific table named "tracks" and the results are appended to a drop-down list. Finally, there is a link to confirm the submis-

Submission (new version)

A submission will consist of a .zip file containing all the source (e.g. .cs) files needed to compile the participant's Rectangle or/and Circle agent(s) and a 2-4 pages description of the agent(s).

Choose the competition in which you want to submit your agent:

track0 ▼

Here you should upload the source code in .zip format.

Upload files

Here you should upload the description of the agent(s) in .pdf format.

Upload files

[Confirm submission](#)

Figure 3. New submission page.

sion, that redirects to a page that notifies the user if the submission has no error. This error will be determined by a automatic script that will run in the server, that will handle new submissions. This part of the error detection belongs to the thesis that the Instituto Superior Técnico student Fábio Almeida is working on. After this first error detection, the idea is that another script will compile and test the submitted AI agents. Since this process takes some time, it is not possible to give an instantaneous feedback. So it was decided that this

Submission Feedback (new)

Successfully submitted!

You will received the results soon in your email account


 Figure 4. New submission feedback page.

information could not be shown in the website, instead an automatic email would be sent to the competitor email address. The final change needed in the website pages is that the results should be compiled and automatically updated in the website results page. In the existing system, it is mandatory that a responsible of Geometry Friends project examines and gather the tests results of the agents performance.

3.2 Developing the Back-End

3.2.1 Organizing Framework Repositories

Development for the Geometry Friends framework began over two years ago. Since then, over four different developers have implemented one or more of its components. These include: the game itself, the competition framework, a level editor and a specific example of an agent implementation. These components were spread over seven different repositories, many of which contained overlapping code, in different stages of development. The team's goal in this area consisted of organizing the latest version of all the components into three different repositories, and marking the old repositories as deprecated. Our first priority was then, the identification of each of the repositories' contents, and their stage of development. We first set out to track changes from the very first version of each component until the latest commit. Subsequently, code overlapping between each repository was pinned down. This allowed us to determine which repositories had the latest version of the components. These were later decoupled, separated, and isolated into the requested three distinct repositories.

3.2.2 Code review and Bug Fixing

With regards to the framework code itself, a major drawback of the previous implementation was that all agents ran in a single thread. This made comparing agent performance harder, since for example, one agent may decide to do nothing during a match. This resulted in the other agent not being able to log in processor time, and it's performance being penalized. A better solution would be to run each agent in its own thread, thereby distributing processor time in a more even manner. The team opted to implement this solution.

In addition to the above, all of the framework code needed to be reviewed, and some nomenclature reevaluated. As already mentioned, the Geometry Friends framework has been in development for over two years, with various people having actively participated in the project. As a result, some code inconsistencies were produced. These were identified and corrected by the team.

3.3 Automatic Testing

Before we started the development, there was no way for the user to get the results of their work in useful time because all of the work needed to compile, execute and run the tests was done by a person, so it could take a while, and since that process was repetitive, there was a great need to automate it. The biggest problem was that the server that received the submission could not run the game because it was a Linux machine and game was made using a framework that only ran on Windows, so we needed to send the code to another machine so we could execute it and then return the results. Several steps needed to be taken on each machine in order to achieve this.

3.3.1 Submission Server

First, we needed to have a mechanism that could watch for submissions via the submission page. We implemented a Java program that watched a specific folder for new entries (user submissions). This Java program is always running in the background like a daemon and listens to events from the File System, that indicates new files or folders, inside the folder

that we chose to watch. We can receive two kinds of files. The first kind is the files from the Submission Web Page or Front-End. Those files are the code submitted by the user, and can not be compiled in the current machine, so need to implement something to send the file to the correct machine so it can compile and run the game, and eventually return the results back to us. This was accomplished by making a small program that receives a file and sends it via FTP (File Transfer Protocol) to the other machine. The second kind of files that we can receive are the result files, that are generated in the Game Testing machine, and indicated the errors and score of the users submitted code. After we identify the result file we need to store the results in a database and send an email to the user for him to check on the results.

3.3.2 Game Testing Server

On this side we needed to start by detecting new files which were transferred from the Submission Server. The same principle from before was applied, a Java program that watches the folders where the files are uploaded to, but with small modifications. When we detect a new submission we decompress it first, so we can access the source code inside, then we compile the source code. If the source code compilation is unsuccessful we return a report to the Submission Server with the compilation error, and if it is successful we proceed with the game execution, where we perform a full test suite (all the levels). After completing all the levels we can finally return the results to the Submission Server again so that they can be processed.

4 RESULTS

4.1 Front-End

Our group managed to make a good beginning for the front-end remaking, since now the files are successfully uploaded to the server's directories directly from the WordPress website and the new pages are ready to give a better and faster feedback to the users. Due to dependencies to incomplete work in the complex area of automatic compilation and AI agents evaluation, it was not possible to make the full

connection to the back-end, but now we have a good skeleton ready to be adapted to the new features. Other existing limitations are that the Wordpress File Upload plug-in doesn't allow to specify the directory that the file should be uploaded to, so in the future there is a need to create a automatic script that organizes the uploaded files to the desired directories. This can be easily done since that in the website page we already connected to the server database to allow to choose the competition that the competitor wants to apply to. Finally, in the future, it is desired that automatic evaluation of the AI agents is made and the results will be written to the database. When the structure of the results in the database is defined, we can easily remake the results page, to show automatically new results and create some kind of ladder-based classification.

4.2 Back-End

4.2.1 Organizing Framework Repositories

The team's efforts in organizing the framework code culminated into the synthesis of all components into three distinct repositories. Namely: the game code, the competition framework and code for a specific advanced game agent. It is our consideration that these endeavours will facilitate further project development.

4.2.2 Code review and Bug Fixing

Our contribution with regards to the agent concurrency problem consisted in the separation of each agent in its own thread. Now, an agent's performance can be better calculated, since it will still act when its peer decides to do nothing at all.

Additionally, the framework code was analyzed and its irregularities identified and corrected. In this way, both competitors and future developers will be able to better comprehend and extend current framework functionality.

4.3 Automatic Testing

4.3.1 Submission Server

We built a successful system to handle the user submissions and send them to a remote machine via FTP. We tested the file transfer

locally and remotely and it always completed without errors. What we didn't test was the result handling because we mismanaged the time it would take to make every single component for this automation process, so we couldn't produce results to both send to the user and to test the database submission.

4.3.2 Game Testing Server

On this machine we successfully built the program that would prepare the user source code for compilation. Unfortunately there was not enough time to make a script that would compile and run the game automatically but we certainly produced the foundation so do that on a near future.

5 CONCLUSION

The framework code base was also adequately separated into a much more manageable set of repositories, which can be understood in a comprehensive manner. Additionally, the entire source code was reviewed and any observed inconsistencies consequently corrected, which will also allow for a better comprehension of the overall project. A central issue regarding the possibility of agent concurrency was corrected as well, and it is our hope this will add fairness to the competitor evaluation process.

Regarding the automation although what was asked a pretty difficult to achieve in so little time, we produced a solid foundation. The functionality that is missing can be easily added and when it's completed, the user submission and testing will be fully automated.

Finally, the front-end is now ready to start communicating with the work done in back-end, in the future it will only be needed to determine the final database structures and how the results should be shown to the client.

ACKNOWLEDGMENTS

We would like to thank Professor Rui Prada that oriented us in this project, and also thank student Fábio Almeida that is working in this project as his thesis and helped us a lot in order to familiarize with the existing project and provided us some resources to work with.

REFERENCES

- [1] GAIPS. (2008) Geometry friends ai competition. IST. [Online]. Available: <http://gaips.inesc-id.pt/gaips/component/gaips/projects/showProject/10/26>
- [2] J. B. Rocha, S. Mascarenhas, and R. Prada, "Game mechanics for cooperative games," *Proceedings of ZON Digital Games 2008*, pp. 72–80.