Botzone: A Game Playing System for Artificial Intelligence Education

Haifeng Zhang, Ge Gao, Wenxin Li, Cheng Zhong, Wenyuan Yu and Cheng Wang
Department of Computer Science, Peking University, Beijing, China

Abstract - *The Botzone Intelligent Agent Platform is an online intelligent agent competition platform. The Botzone is designed to evaluate the intelligence of agents through competitions between them. The Botzone provides a context for users to learn basic AI knowledge and apply a variety of AI algorithms. It has been used as a teaching system and has served National Olympiad in Informatics (NOI) as a competition platform for many years. The Botzone is also an expandable system, which provides a set of facilities and clean interfaces, allowing users to design different competing situations and evaluate differing types of intelligent agents. In this paper, we introduce the structure, the applications, and the expansibility of the Botzone Intelligent Agent Platform.*

Keywords: Intelligent Agent, Online System, Education

1 Introduction

The potential for computer games as a tool for AI research and education continues to blossom [1]. Well-designed Games need great intelligence and good strategy. Designing an agent that plays the game is a challenging task, thus it provides an ideal context for practice of AI algorithms. Many game-based tools for CS and AI educators to use have been presented so far. Mcgovern et al. and Denero et al. have proposed several Java-Based games such as Space-war and Pac-man which can be used to teach introductory Artificial Intelligence [2, 3]. However, they don't provide a single platform for managing the various games used in CS education. Users can only develop and test their game agents offline and it is difficult for users to compare their agents with others’. Furthermore, those tools provide no uniform set of interfaces for educators to design different games to teach different algorithms.

The Botzone Intelligent Agent Platform is designed as an online turn-based strategy game playing system. Students major in Computer Science can create game agents and participate in the contest with others on the platform. In the process, they can learn basic programming skills and many Artificial Intelligence algorithms. Currently, such platforms for educators and researchers to use for turn-based strategy game development are limited. MGPS [4] is a similar platform, but it only supports normal-form games, which are only a small part of the turn-based strategy game.

A turn-based strategy game is a strategy game where players take turns when playing, distinguished from real time strategy where all players play simultaneously [5]. This type of game is commonly seen (such as chess and bridge), so it is easy to be understood by students. Furthermore, according to the complexity of the games, they can be applied in various courses. For example, the simplest Gobang game can be used in introductory classes while the Robocup game [6] can be used in advanced artificial intelligence classes. So the platform is suitable for students from undergraduates to graduates. We also provides a set of facilities and clean interfaces, allowing users to design different competing situations and evaluate differing types of intelligent agents.

We have applied the Botzone in an introductory programming course and an advanced algorithm course. We developed two turn-based strategy games for these two courses respectively. We try to lead students to experience the entertainment of programming through the Botzone. As a result, the students are more interested in such a study method, rather than in the traditional ones. In addition, we have applied the Botzone in a large-scale live competition. The competitors develop agents for an originally designed game. With the help of the Botzone, it is convenient for the organizer of the competition to select those competitors who have deeper understanding in applying AI algorithms.

We first review the structure and discuss the features of the Botzone. Then we present the main games on the Botzone and the applications of them in courses and competitions. Finally, we introduce the interfaces for developing new games, and invite users to develop new games or request for educational cooperation.

2 System Overview

The Botzone Intelligent Agent Platform is an online network system, which is now available for public use on our servers. Users may register accounts on the website and log in. Then they can view the description and the programming interfaces of the game. Once users have finished developing their agents, they can upload the source code of the agents using a simple web form. After that, the users can participate in the contests held on the platform with their own agents. They can also just start a single match with other agents on the platform.

A *match* is a single game process which is played by several agents. The agents take actions by turns until the game is over. The result of the game can reflect relative intelligent levels of the agents. A *contest* is a set of organized matches. It is participated in by a set of agents. The objective of a contest is to rank the agents by the relative intelligent levels. The matches of a contest can be organized in different competition systems, such as round robin[1] and Swiss system[2]. A series of contests can be established for a long-term project, such as a full-semester course or a large-scale competition.

The Botzone can be divided into three parts: the frontend, the storage system and the judge system. The relationship among the three parts is shown in the Figure 1.

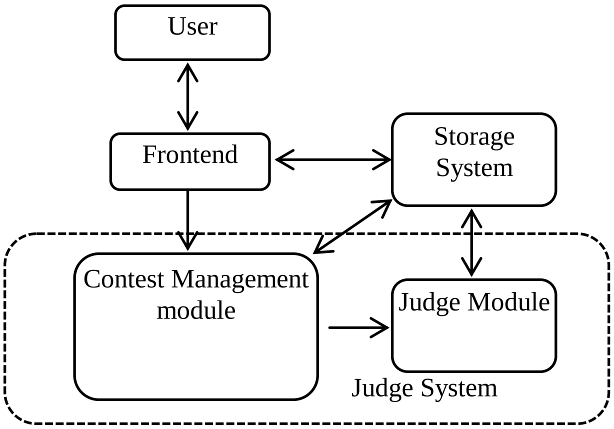

Figure 1: The structure of the Botzone System.

The frontend mainly refers to the Botzone website. Most of the interactions with users are completed by this part, such as registering accounts, uploading agents, attending contests, viewing results etc. The storage system consists of the storage of the executable files of the agents, logs, and the database storing information about users, agents and contests. The judge system consists of the contest management module and the judge module. The contest management module can run a contest according to a certain competition system. It decides the players of each single match, and then calls the judge module to run the match and record the logs, including the winner, the scores and the process of the match.

The Botzone has supported several functions which make it more entertaining as well as educational. First, it provides a visualized way to view the match process other than log files. There are FLASH animations for each game, which can replay the process of a certain match according to the log files. We find that users of the Botzone spend over half of the time on watching the FLASH animations. According to the feedbacks of the students, it also makes them more interested in developing agents and participating in the contests. Second, we allow users to choose their opponents among thousands of the agents on our platform. In this way, users can learn AI strategies from others' agents and improve their own algorithms. We have also provided a privacy function to protect users' agents. If one would not like to share his agents, he can set the type of the agent to be private and then the agent is no longer available to be chosen as opponent by others.

3 Applications

The Botzone now have three main games which are available for public use, Gobang Game, Blokus Game[3] and FourColours Game.

Gobang (Figure 2) is a simple and well-known abstract strategy board game:

> It is traditionally played with go pieces (black and white stones) on a go board (19x19 intersections). Black plays first, and players alternate in placing a stone of their color on an empty intersection. The winner is the first player to get an unbroken row of five stones horizontally, vertically, or diagonally. [7]

[1] Round-robin system is a competition organization method that gathers every n (defined by game rule) agents to play a match and total up the score of every agent.

[2] Swiss system is also a competition organization method that arranges several rounds. In every round an agent plays a match with its neighbor(s) in the current rank list, expecting the rank list is approaching stable.

[3] Blokus (pronounced like "block us") is an abstract strategy board game for two to four players, invented by Bernard Tavitian and first released in 2000 by Sekkoïa, a French company.[8]

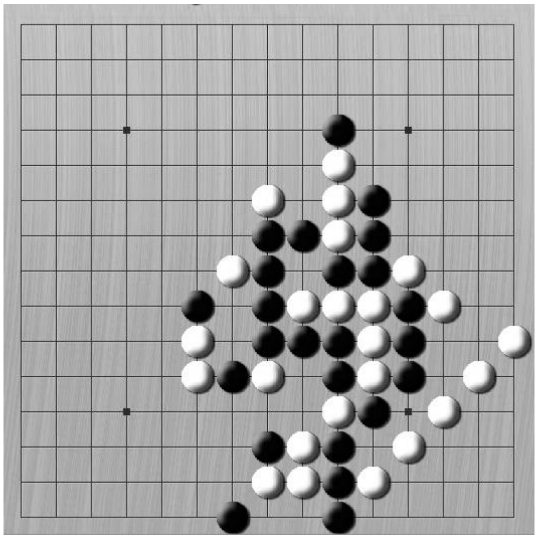

Figure 2: An example of Gobang game. In this game, the White wins.

Blokus (Figure 3) is a little more complex. The rules of Blokus are as follows [8]:

- This is a four-player game. Each player owns a set of prescriptive blocks with a unique color.
- Order of play is based on color, with blue going first, followed by yellow, red, and green.
- The first piece played of each color is placed in one of the board's four corners. Each new piece played must be placed so that it touches at least one piece of the same color, with only corner-to-corner contact allowed—edges cannot touch. However, edge-to-edge contact is allowed when two pieces of different color are involved.
- When a player cannot place a piece, he or she passes, and play continues as normal. The game ends when no one can place a piece. Result is decided by the sum of area of each player's blocks.

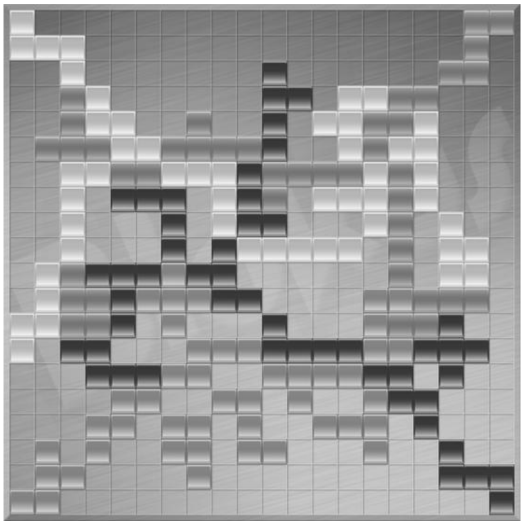

Figure 3: An example of Blokus game.

The game FourColours (Figure 4) was originally designed for National Olympiad in Informatics (NOI) 2010 and 2011. The idea of FourColours derives from the famous four color theorem. The basic rules are as follows:

- This is a game for two players. One controls Color 1 and Color 3, while the other controls Color 2 and Color 4. Color 1 goes first, followed by Color 2, 3 and 4.
- The two players paint a randomly generated planar map in turns. The map is consisted of 30-50 areas, which vary in both shape and size.
- The areas are of three types: lands, seas and mountains. The player can only paint lands. The color of a sea area is the color which appears most in the adjacent land areas. The color of a mountain area is the color which appears latest in the adjacent land areas.
- The adjacent areas can't be painted with the same color.
- The game ends when no more area can be painted. The one who paints more total size of areas wins the game.

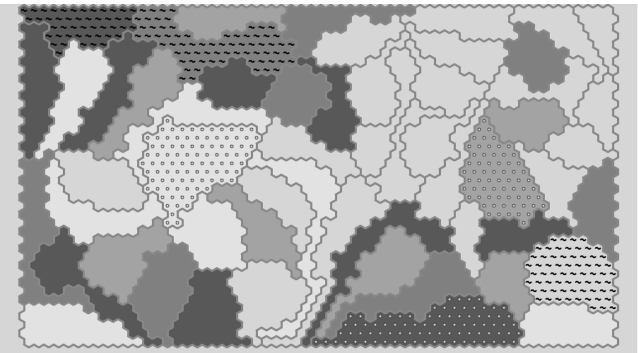

Figure 4: An example of FourColours game. Areas with tildes represent seas and areas with circles represent mountains.

These three games are all well suited to be used as a supplement to the Artificial Intelligence course or competition. They provide an ideal context for students to learn various search and pruning algorithms, especially min-max search with alpha-beta pruning, which is among the strongest algorithms in such problems [9]. Students can see the benefits and the necessity of pruning, because the game tree is too huge. And we find that the implementing of such game agents can deepen students' understanding of the search and prune algorithms through the final exams and the questionnaire of our courses.

Furthermore, we find that playing games is also a good way for students to explore some learning algorithms. The interaction of machine learning and game playing has long been a topic since the early days of Artificial Intelligence [1]. It is difficult to come up with an ideal evaluation functions to evaluate each situation immediately. However, we can analyze the result and the log of hundreds or even thousands of matches and then get the knowledge of each situation and its outcome to improve our evaluation functions. Some possible approaches include decision trees, clustering etc.

The platform has been used in our courses for many years. We encouraged students to use all kinds of algorithms in the projects as long as the algorithm belongs to the field of Artificial Intelligence. We run a series of contests, and students are quite interested in such a project. In the advanced AI course last year, over 60% of the students kept on improving their algorithms throughout the whole semester and we have found many creative ideas. They enjoyed the assignment on the platform while learning the Artificial Intelligence. What's more, those who perform well in the project often get a high score in the related questions in the final exams.

In addition to being applied on courses, the Botzone has served NOI as a competition platform for three years. NOI is a nation-wide annual competition, aiming to improve high school students' programming skills and select members of national team to attend International Olympiad in Informatics (IOI). The Botzone helps NOI to examine competitors' programming skills and their familiarity with the related algorithms. The competition can also stimulate their interests in exploring the world of artificial intelligence further.

4 Expansibility

Since the beginning of developing our system, we have realized the importance of the expansibility. Only the few games on the platform can't satisfy the demand of various courses and competitions. The students will soon feel bored and lose interest in attending the courses and the competitions. In order to collect more interesting and educational games on the Botzone, we open up a new game interface to the public, hoping to make the platform more attractive. With the increase of the number of all kinds of games, students can learn more artificial intelligence algorithms and learn how to apply them in different games, since different games require different strategies and algorithms.

Users can design and develop new games by themselves and upload it to the Botzone. A new game needs the following files to be run well on the server:

Judge. Committers should design the rule of the new game, and implement the judge module. The judge module runs a match, maintains game process status, sends commands to agents, and outputs results. The judge module communicates with agents through controller, a built-in module in the Botzone, which plays a role as a messenger between the judge module and agents. The controller module interacts with the judge module and agents by command-line. The protocol between the controller module and the judge module is shown in the Table 1.

Display. Committers should provide a display module of the new game. The Botzone only support FLASH animations at present. The display module is in charge of displaying a single match, which is finished and has been logged on the Botzone server. The display module fetches match information from the Botzone server through HTTP request when it is played in a user's browser. Therefore, the process of every match can be replayed visually.

Introduction. Committers should provide an introduction of the new game. This introduction is prepared for the agent-developers of the new game. It should contain detailed explanation of the rule of the new game and the communication protocol between the judge module and agents.

Sample. Committers should provide a copy of source code of a sample agent for the new game. This sample is also prepared for the agent-developers. It could be equipped with a basic intelligent level, but it should cover all the cases on the communication protocol.

Table 1: Communication Protocol between the controller module and the judge module

	Instruction	Function
Controller To Judge	[PLAYERNUMBER] number	Tell Judge the number of players
	[PLAYERNAMES] name1 name2…	Tell Judge the names of the players
	[START]	Notify Judge that game started
	[TIMEOUT]	Tell Judge that the last agent got a time-out error
	[NEXT] pre_command	Feedback the output of the last agent
Judge To Controller	[PLAYER] player_num	Appoint the next agent
	[COMMAND] command	Send command to the next agent
	[KILL] player1 player2…	Stop one or more agents
	[GAMEOVER]	Notify Controller that the game is over
	[RESULT] score1 score2…	Output results
	[VIEWLOG] one line log	Output log file for the display module

We will review these files after committers upload them. The new games that are clearly defined, moderately challenging and fairly interesting will be accepted. In this way, we can expand the application of our platform.

Recently, we have accepted a new game named Snake (Figure 5), which is developed by an ordinary user. The main rules of Snake are as follows (similar to the Snake game in mobile phones):

- This is a game for two players. Each player controls a snake.
- The snakes can move ahead, turn left or right in every turn. It will die if it moves out of the map, or hits the body of either himself or the opponent.
- There are some stars coming out in succession, which may be eaten by snakes. When a snake eats a star, it will lengthen its body and gain a point. The longer, the better.

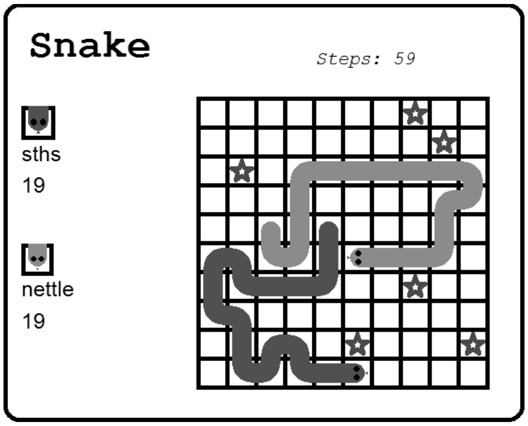

Figure 5: An example of Snake game.

With the increase of the variety of games on the Botzone, it can be applied in more courses such as game theory class or machine learning class, and the students can learn more related artificial intelligence algorithms and ideas.

5 Conclusions

The Botzone is an educational and entertained online game playing system. It can be used to teach students basic programming skills and algorithms in the AI field. It also can be applied to hold large-scale competitions. What's more, it is an expansible system so that it can provide services in more situations and for more users. In the recent years, the platform has been successfully applied in various courses and competitions.

The platform is now available for public use on the internet. We are eager for people to design new games for us and use our platform in courses and competitions. Those interested parties may contact the lead author with such requests.

In the future, we will continue improving our system and collect more excellent games developed by users. We will also consider developing the function that enables users to play game with the agents.

6 References

[1] Bowling, M.; Furnkranz, J.; Graepel, T.; and Musick, R.. "Machine learning and games". Machine Learning 63:211–21, 2006.

[2] Mcgovern, A.; Tidwell, Z.; and Rushing, D.. "Teaching Introductory Artificial Intelligence through Java-Based Games". AAAI Symposium on Educational Advances in Artificial Intelligence, North America, 2011.

[3] DeNero, J.; and Klein, D.. "Teaching introductory artificial intelligence with Pac-man". Educational Advances in Artificial Intelligence (EAAI), 2010.

[4] Macglashan, J.; Miner, D.; and Desjardins, M.. "A Game Playing System for Use in Computer Science Education". Florida Artificial Intelligence Research Society Conference, North America, 2010.

[5] Wikipedia. Turn-based strategy—Wikipedia, the free encyclopedia. [Online; accessed 7-February-2012].

[6] Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawai, E.; and Matsubara, H.. "RoboCup: A Challenge Problem for AI and Robotics". RoboCup, pp. 1-19, 1997.

[7] Wikipedia. Gomoku—Wikipedia, the free encyclopedia. [Online; accessed 7-February-2012].

[8] Wikipedia. Blokus—Wikipedia, the free encyclopedia. [Online; accessed 7-February-2012].

[9] Allis, L. V.; Herik, H. J. V. D.; and Huntjens , M. P. H.. Go-Moku Solved by New Search Techniques. Computational Intelligence - CI 12(1): 7-23, 1996.