Bot!Battle! Projects – Spring 2016

Bot!Battle! system is a web-based system that facilitates computer science education through programming challenges in which users write simple programs that control a player's action in an animated game.

Users will be able to:

- Register for the system,
- Choose a game challenge from various categories, learn prerequisite information needed for the challenge, upload and test bot programs, publish their bot so that it is accessible to others, and run games against other users published bots. Initially the only category that will exist will be *Introduction to Java*.
- Enter a contest, which will consist of one game challenge, and enter a bot in the contest.
- View the results of a game or a contest (either live or one that has already taken place).

Languages/Platforms/Databases Used

Currently it is envisioned that the web server applications will be developed in Javascript using NodeJS and the JADE template engine.

The client side will use the AngularJS framework for most pages and the Phaser game engine for the graphical depiction of games.

The Bot Evaluation System will be written using Java. Since the bot programs and game evaluation programs will be uploaded by users, these will be run in the Isolate sandbox system (https://github.com/cms-dev/isolate).

The various systems that comprise Bot!Battle! will be integrated via a MySQL relational database.

Bot!Battle! Web Server Application

The Bot!Battle! web server application will provide the following functionality:

- A landing page allowing the user to login or register
- Once logged in, a page that includes the following items:
    - Lessons, with a list of all of the lessons, grouped by category
    - Challenges, with a list of the game challenges, grouped by category
    - Contests, with a list of contests
    - Admin link (if the user is an administrator)
- Selecting a lesson from the lessons category will display a page with text and an optional video (this content will be provided by Dr. Blum).
- Selecting a challenge will display the information required to write a bot to complete the challenge, links to relevant lesson pages, links to testing arena where a bot can be tested, and the ability to publish a bot (i.e. make it public).
- Selecting a contest, will allow users to upload a bot for a contest, and view results of in-progress or completed contests.
- The admin link will allow an administrator to add, edit or delete challenges or contests. For contest administration, the administrator will be able to not only specify parameters for the contest, they will also be able to control the flow of the contest, e.g. running an entire round of a contest in parallel, or choosing to run individual games in series, with the games being broadcast.

Note that the content for all of the dynamic pages will be pulled from the MySQL database.

This group will also produce a challenge for use in the system, in the form of a challenge description for bot development, a game evaluation engine, and resources used in the visualization of the games.

Game Display, Testing Arena, and a Game Challenge

In addition to the evaluation system, this team will be responsible for creating the structure for the Game Display Module. This Game Display Module will provide the framework for challenges to be defined within the system. This module will include the formats for bots, the challenge evaluation engines, and the visualization of the games.

The bots will be designed to play a single turn in a game at a time. They will take as input the current game state and produce output that indicates their next move.

The game display module will allow for two modes:

1) A playback mode, which allows users to play, pause, rewind, and fast-forward a game visualization. Note that for live games, these controls should be disabled.
2) A testing mode, which allows users to tweak their code as they play a game. If the game challenge is a two-player challenge, they will be able to choose either a public bot to play against, or they can enter commands themselves. The testing mode will allow them to rewind to a previous turn.

This group will also produce a challenge for use in the system, in the form of a challenge description for bot development, a game evaluation engine, and resources used in the visualization of the games.

Bot Evaluation System and Game Challenge

The Bot Evaluation System is a Java program that runs games that have been queued up as indicated by the MySQL database system. There are three modes under which a game will be run:

Testing Mode

In testing mode, the system will run a single move, given a current game state, the user (or bots) move.

Game Mode

In game mode, the system will run an entire game to completion.

Contest Mode

Contests will start at a designated time, and will run in two stages. In stage 1, entered bots will play an admin-designated number of games, and be assigned a score, based on an approach similar to the Elo rating system (or some other chess player rating system). The rating system will include a way to break ties.

The administrator will then be able to designate that a certain number of the top teams will play in a single elimination tournament. The number must be a power of two, and the teams will be seeded in the tournament, such that the top ranked player plays the weakest player in the tournament, the second top player plays the second weakest player, etc.

This group will also produce a challenge for use in the system, in the form of a challenge description for bot development, a game evaluation engine, and resources used in the visualization of the games.

The game evaluation engine will be designed to do one of the following, based on its input:

1) Create an initial game state
2) Take as input a current game state and a player move, and determine:
    a. if the move is legal,
    b. whether one player or the other has won the game,
    c. the next game state, and
    d. the commands to visualize the move that was made.