

The project is a turn-based educational game that pits java bots (uploaded by the users) against each other or single-bot challenges. Our part of the project deals with the testing arena, as well as playback mode and the graphical display.

## Requirements

1. The Game Display Module must contain a library of built-in graphical assets including sprites, backgrounds, and animations.
2. The Game Display Module must be able to initialize the display of objects, sprites, and text entities.
3. The Game Display Module must be able to accept a Game Initialization Message including a background, default timestep, and list of entities, and initialize the game display on the Game Display Window using this data.
4. The Game Display Module must be able to modify the Status Table on the Playback Mode Webpage or Testing Mode Webpage, including the Turn, Standard Input, Standard Output, and Standard Error columns.
5. The Game Display Module must be able to receive a message from the server including a set of changes to execute for the next turn, or for every turn.
6. The Game Display Module must be able to display and animate a list of changes for any given turn using the Game Display Window.
7. The Playback Mode Webpage must be able to accept a request for a specific match and poll the server until the data for the requested match is available.
8. The Playback Mode Webpage must be able to display the Game Display Window, the Playback Mode Controls, and the Status Table.
9. The Playback Mode Controls must include Play, Pause, Go Back One Turn, Go Forward One Turn, Go To Start, Go To End, and Change Speed controls to adjust replays of finished matches.
10. The Testing Arena Webpage must be able to display the Game Display Tab (including the Game Display Window), the Status Table, and a Player Tab for each player.
11. The Testing Arena Webpage must allow for any player in a Testing Arena match to upload a java file to use as their bot.
12. The Testing Arena Webpage must provide a built-in IDE text editor allowing any player in a Testing Arena match to write a java program, which can be uploaded and used as their bot.
13. The Testing Arena Webpage must allow for any player in a Testing Arena match to choose a public bot stored in the Bot Database to use as their bot.
14. The Testing Arena Webpage must provide a list of default bots stored in the Bot Database and allow for any player in a Testing Arena match to choose one such bot to use.
15. The Testing Arena Webpage must be able to send a message to the server indicating that it should begin the next turn.
16. The Testing Arena Webpage must be able to poll the server until requested data for the next turn is available.
17. The Server must be able to poll the Game State Database until requested data is available.
18. The Server must be able to extract the full match data of a completed match from the Game State Database and send it to the Playback Mode Webpage.

19. The Server must be able to send a Game Initialization Message to the Testing Arena Webpage.
20. The Server must be able to send a list of changes for the next turn to the Testing Arena Webpage.
21. The Server must be able to accept an uploaded java file and store it in the Bot Database.
22. The Server must be able to accept an uploaded string, write it to a java file, and store it in the Bot Database.
23. The Server must be able to verify that a bot with a given ID exists in the Bot Database.
24. The Server must be able to send confirmation and error messages to the Testing Arena Webpage or Playback Mode Webpage.
25. At least one playable game challenge must be available (created by our team).

## Glossary:

**Bot:** A java program stored in the Bot Database which takes a game state and returns an input command. The Testing Arena can be used to test bots.

**Public Bot:** A bot created by a player which has been made publicly available, allowing other players to use it.

**Private Bot:** A bot created by a player which has been made private, preventing other players from using it.

**Preloaded Bot:** A default public bot created by the Bot!Battle! programmers.

**Challenge:** A game mode run on the server that is played by one or more bots. Upon completion of a challenge mode match, users can view it in playback mode (using data stored in the Game State Database).

**Change:** A command associated with a particular turn which animates or changes the properties of an entity.

**Default Timestep:** The default length of every turn, in seconds.

**Entity:** The graphical representation of an object on screen such as a player, a wall, or a flag.

**Sprite (Entity):** An entity which has associated animations and can accept changes.

**Object (Entity):** A still, unmoving image which does not accept changes.

**Text Entity:** A text string drawn on the screen which can accept changes.

**Game Display Module:** The code which implements Phaser functionality to draw images on the Game Display Window.

**Game Display Tab:** A tab on the Testing Mode Webpage which displays the Game Display Window.

**Game Display Window:** A HTML canvas that the Game Display Module draws on, which appears on the Testing Mode and Playback Mode Webpages.

**Game Initialization Message:** A message which contains a background, default timestep, and list of entities for initializing the initial game state.

**Game State:** The state of a match at the beginning of a turn, consisting of the position and state of all entities.

**Match:** An instance of Bot!Battle! being played in the Testing Arena or replayed in Playback Mode.

Phaser: A HTML5 game framework used by the Game Display Module.

Playback Mode: A game mode designed for viewing replays of finished matches.

Playback Mode Webpage: A webpage displaying the Game Display Window, the Playback Mode Controls, and the Status Table.

Playback Mode Controls: Buttons displayed on the Playback Mode Webpage used to adjust replays of finished matches, including Play, Pause, Go Back One Turn, Go Forward One Turn, Go To Start, Go To End, and Change Speed.

Player Tab: A menu on the Testing Arena Webpage which displays options for bots a player can use (Upload File, Edit Code, Public Bot, or Preloaded Bot).

Server: Software hosted on a remote machine used for communication between the webpages, databases, and various external functionalities.

Status Table: A table of text output including the Turn, Standard Input, Standard Output, and Standard Error columns.

Testing Arena: A game mode designed for bot testing.

Testing Arena Webpage: A webpage displaying the Game Display Tab (including the Game Display Window), the Status Table, and a Player Tab for each player in the current match.

Turn: A game state and associated changes.