**Functional Requirements**

-The client shall accept as input a game state (as well as commands from server-side hosted bots) from the server and display it using the game display module.

-The client shall provide the ability to replay a finished game using a sequence of commands and game states passed to it by the server.

-The game display module shall display the current game state with sprites, using data passed to it from the client.

-The client shall accept as input commands from the users and/or client-side java bots and pass it to the server.

-The client shall accept as input a java bot from the user and will upload it to the server to produce input for testing arena games.

-The server shall accept a command from the client and pass it to the game evaluation engine. If the command is accepted, the server shall receive the next game state from the engine and pass it to the client(s). If not, the client responsible for the invalid data will notify the user of the problem.

-If the client requests a replay of a game, the server will pass it all game states and commands recorded for the requested game.

-The server shall store the current and previous states of all live and completed games in the game state database.

-The server shall accept uploaded java bots and ensure that they are not malicious. If the uploaded bots are unsafe, the system will send a message to the client to inform the user of the error.

-If a game being played involves one or more uploaded bots, the server shall run those bots to produce the needed commands.

**Glossary**

**System:** All components of the project, including the client and server-side modules.

**Client:** The program used by the end-user.

**Server:** The software that runs on the

**Game State Database**: The database that stores the states of each game for use in game evaluation and testing.

**Game Display Module:** Client-side Phaser program which takes game input from the server and displays it in graphical form.

**Character:** The in-game representation of a player or bot.

**Bot:** A java program that takes a game state and returns an action.

**Public Bot:** A default bot provided by the server, used for testing.

**Player Bot:** A player-created bot.

**Command:** The move a character makes on their turn, which changes the game state.

**Testing Arena:** Game mode which allows the user to test bots against other bots (hosted server-side) or player input.

**Challenge:** An online game mode that accepts only bot input.

**Turn:** A single game state update**.**

**Game:** An instance of testing or challenge mode.

**Live Game:** A game that is still currently being played on the server.

**Mode:** Testing, challenge, or playback.

**User:** Human who plays the game.

**Testing Mode:** A game mode which allows the user to test bots in the testing arena.

**Playback Mode:** A game mode which allows the user to view a game that has already been played.

**Play:** Start or resume playback of a game.

**Pause:** Stops playback of the current game, with the ability to resume later.

**Rewind:** Reverts the game state back in the visualization.

**Fast-Forward:** Speeds up the display of the game visualization.