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 testing mode and graphical display.

The project has three major game modes: testing mode (in which the user can test their bots against other bots, user input, or single-player challenges), challenge mode (in which the server will run a match using only an uploaded bot or bots), and playback mode (in which the Game Display Module will replay a finished testing mode or challenge mode game using data retrieved from the server).

**Game Display Module:**

1. The Game Display Module must be able to draw game states and animations, including the map, characters, and props. Additionally, the module must be able to display notifications.

2. The Game Display Module must be able to accept animation commands, and notifications from the server for specific matches (either individually or in a sequence).

3. The Game Display Module must be able to accept an initial game state from the server in the beginning of a match.

4. The Game Display Module must provide functionality for playback mode, allowing the user to replay a completed match one turn at a time, as well as rewind it (play the game animations backwards) or fast-forward it (play the game animations at a faster speed).

**Testing Mode:**

5. The controller must be able to upload java bots to the server for use in testing mode.

6. At the beginning of any character’s turn, the controller must be able to handle input from them as appropriate (based on their given input type).

a. If the character is controlled by user input, the controller must be able to prompt the user for their input and then send it to the server, then end the character’s turn and wait for the server’s response.

b. If the character is controlled by bot, the controller must be able to prompt the user to reupload their bot, if desired. The controller must be also be able to prompt the user to proceed to the next turn. If the user accepts, the controller must send a message to the server indicating this and then wait for the server’s response.

7. When prompting the user for input at the beginning of a character’s turn, the controller must also allow for the user to undo or redo previous moves.

**Challenge Mode:**

8. At least one playable challenge mode will be available (created by our team).

**Glossary**

**Animation Command:** A message sent to the Game Display Module indicating that it should cause a specific character to display a specific animation (and move to a specific tile if needed).

**Bot:** A java program hosted on the server that takes a game state and returns an input command. Users can upload bots to the server. Testing mode can also be used to test bots.

**Public Bot:** A default bot provided by the server.

**Player Bot:** A bot created by a player.

**Challenge:** An online game mode run by 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).

**Character:** The in-game representation of a player or bot. A character is controlled by input commands (produced by a player or bot). The Game Display Module draws the character’s sprite, and animates it when passed animation commands.

**Controller:** (Short for Testing Mode Controller) A module embedded in the webpage which runs testing mode.

**Fast-Forward:** In playback mode, speeds up the game animations.

**Game Challenge:** A type of challenge in challenge mode with a specified win condition and number of players.

**Game State:** The map a match is being played on, as well as positions and states of all characters and props on it at the beginning of a turn or at the end of the match.

**Game State Database**: A server database that stores all game states, animation commands, and notifications for all matches.

**Game Display Module:** A module written in Phaser which takes game states, animation commands, and notifications, and displays them in the Game Display Window. The Game Display Module also runs playback mode, allowing the player to view replays of finished matches.

**Game Display Window:** A window in the web application that displays the Game Display Module to the user.

**Input Command:** Data produced by a player or bot that indicates a desired action for a character to take.

**Map:** An arena in which testing or challenge mode takes place. A map contains walls, empty space, characters, and other props.

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

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

**Notification:** A message to display using the Game Display Modul. Will inform the user if the match has run to completion (and the result of the finished match), or if an error has occurred.

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

**Pause:** In playback mode, temporarily stops playback of the match being watched.

**Play:** In playback mode, starts or resumes playback of the match being watched.

**Playback Mode:** A game mode which allows the user to view a replay of a testing mode or challenge mode match that has already been played. The match info (game states, animation commands, and notifications) will be passed to the Game Display Module by the server when the user requests a particular game.

**Prop:** An object on a map that is neither a wall nor a character, such as a flag or moving car. Props must be able to accept animation commands.

**Rewind:** In playback mode, replays the match being watched in reverse.

**Server:** Software that is hosted on a remote machine, including databases.

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

**Testing Mode:** A game mode designed to allow the user to test their bots, controlled by the Controller. Testing mode will allow the user to specify a type of input for each character (user or bot input). Testing mode will not run automatically like challenge mode but instead after each turn will prompt the user to provide input (if the current character is user-controlled) or move to the next turn (if the current character is bot-controlled). Testing mode will also allow the user to undo or redo moves.

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

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