USE CASE: USER INPUT IN TESTING MODE

**BASIC COURSE:**

The game webpage displays an input field for a command, a “submit” button, and an undo button. The user enters a command and presses the submit button. The Mode Controller reads the input from the webpage. The Testing Mode Controller sends a message to the server including the user’s input command and turn number. The server then returns the game display data (a sequence of animation commands and notifications corresponding to the bot’s turn) to the Game Display Module. The Game Display Module passes the game display data to the Game Display Window. The Game Display Window in turn displays the game animations to the user. The Game Display Module then saves the game state. The Testing Mode Controller then increments its turn number.

**ALTERNATE COURSES:**

**Current turn is the first turn of the game, the user has not undone any turns:** The game webpage displays an input field for a command, and a “submit” button.

**Current turn is the first turn of the game, the user has undone at least one turn:** The game webpage displays an input field for a command, a “submit” button, and an redo button.

**Current turn is a previous turn which has been undone, but not the first turn:** The game webpage displays an input field for a command, a “submit” button, and an undo button.

**User presses undo button:** The Game Display Module starts the previous turn paused.

**User presses redo button:** The Game Display Module starts the previous turn paused.

**The game has ended:** The Game Webpage displays only the undo button.

USE CASE: BOT INPUT IN TESTING MODE

**BASIC COURSE:**

The game webpage displays a Run Bot button, a Reupload Bot button, and an undo button. The user presses Run Bot. The Testing Mode Controller reads the input from the webpage. The Testing Mode Controller sends a message to the server including the user’s request and turn number. The server then returns the game display data (a sequence of animation commands and notifications corresponding to the bot’s turn) to the Game Display Module. The Game Display Module passes the game display data to the Game Display Window. The Game Display Window in turn displays the game animations to the user. The Game Display Module saves the game state after receiving the game display data from the server. The Game Display Module then signals the Testing Mode Controller to begin the next turn.

**ALTERNATE COURSES:**

**Current turn is the first turn of the game, the user has not undone any turns:** The game webpage displays a Run Bot button and Reupload Bot button.

**Current turn is the first turn of the game, the user has undone at least one turn:** The game webpage displays a Run Bot button, Reupload Bot button, and Redo button.

**Current turn is a previous turn which has been undone, but not the first turn:** The game webpage displays a Run Bot button, Reupload Bot button, Redo button, and Undo button.

**User presses undo button:** The Game Display Module starts the previous turn paused.

**User presses redo button:** The Game Display Module starts the previous turn paused.

**The game has ended:** The game webpage displays only the undo button.

**The user presses Reupload Bot:** The Game Webpage enables the bot uploader (external functionality) allowing the user to reupload a bot. The Game Webpage then freezes all input. The bot uploader then returns the result. The Game Webpage then unfreezes input and displays the result of the attempted upload.

USE CASE: ENTERING PLAYBACK MODE

**BASIC COURSE:**

The game webpage displays a Match ID input field and a Submit button. The user enters the Match ID of the match to retrieve and clicks submit. The Game Webpage confirms that the user entered a valid Match ID format. The Game Display Module sends the Match ID to the server. The server returns the game display data to be viewed (game states, commands, and notifications) to the Game Display Module. The Game Display Module passes the game display data to the Game Display Window. The Game Display Window in turn displays the game display data for the initial state to the user.

**ALTERNATE COURSES:**

**Invalid Game ID Format:** The Game Webpage will display an error.

**Game ID does not exist:** The server will return an error indicating that the game could not be found to the Game Display Module. The Game Display Module then passes the error to the Game Webpage. The Game Webpage in turn displays the error.

USE CASE: PLAYBACK MODE PAUSED

**BASIC COURSE:**

The Game Display Module starts a paused turn. The Game Display Module then passes its game display data to the Game Display Window. The Game Display Module also signals the Game Webpage to display the playback mode buttons. The Game Display Window displays a paused turn in playback mode. The Game Webpage then displays Rewind, Play, and Fast-Forward buttons. The user presses the Play Button. The Game Webpage signals the Game Display Module to start animating the turn. The Game Display Module passes the game display data to the Game Display Window. The Game Display Window in turn displays the game animations to the user. The Game Display Module also signals the Game Webpage to replace the pause button. The Game Webpage then complies and replaces the pause button with the play button.

**ALTERNATE COURSES:**

**User presses the Fast-Forward button:** The Game Webpage signals the Game Display Module to start the next turn, paused. The Game Display Module passes the game display data for the requested turn to the Game Display Window. The Game Display Window in turn displays the turn to the user.

**User presses the Rewind button:** The Game Webpage signals the Game Display Module to start the previous turn, paused. The Game Display Module passes the game display data for the requested turn to the Game Display Window. The Game Display Window in turn displays the turn to the user.

**Current game state is final game state:** The Game Display Module displays Rewind and Fast-Forward button (but not Play or Pause button).

USE CASE: PLAYBACK MODE PLAYING

**BASIC COURSE:**

The Game Display Module starts animating a new turn. The Game Display Module sends the game display data associated with the current animations to the Game Display Window. The Game Display Window then displays the game animations. The Game Display Module then ends the turn. The Game Display Module starts animating the next.

**ALTERNATE COURSES:**

**User presses the Pause button:** The Game Display Module starts the turn in a paused state. The Game Display Module then signals the Game Webpage to replace the Play button with the Pause button.

**User presses the Fast-Forward button:** The Game Display Module increases the game animation speed. The Game Webpage depresses the Fast-Forward button and releases the Rewind button.

**User presses the Rewind button:** The Game Display Module reverses the game animation speed. The Game Webpage depresses the Rewind button and releases the Fast-Forward button.

**User releases the Rewind button:** The Game Display Module restores the game animation speed to its normal rate.

**User releases the Fast-Forward button:** The Game Display Module restores the game animation speed to its normal rate.

**Current game state is final game state:** The Game Display Module signals the end of animation to the Game Display Window.