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 clicks the submit button. The game webpage then sends the user input to the Testing Mode Controller. The Testing Mode Controller sends a message to the server including the user’s input command. 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 display data 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 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 clicks undo button:** The Game Display Module retrieves the stored data from the previous turn and sends it to the Game Display Window to display.

**User clicks redo button:** The Game Display Module retrieves the stored data from the next turn and sends it to the Game Display Window to display.

**The game has ended:** The Testing Mode Controller will refuse any further input from the user, except for undo.

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 clicks “Run Bot”. The game webpage then sends the user’s request to the Testing Mode Controller. The Testing Mode Controller sends a message to the server including the user’s request. 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 display data 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 clicks undo button:** The Game Display Module retrieves the stored data from the previous turn and sends it to the Game Display Window to display.

**User clicks redo button:** The Game Display Module retrieves the stored data from the next turn and sends it to the Game Display Window to display.

**The game has ended:** The Testing Mode Controller will refuse any further input from the user, except for undo.

**The user clicks 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 and waits for a response from the bot uploader indicating the upload was either successful or unsuccessful. 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 specifies the Match ID of the match to retrieve and clicks submit. The Game Webpage confirms that the user entered a valid Match ID. The Game Display Module sends the Match ID to the server. The server returns the game 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 game state. 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 game state in playback mode. The Game Webpage then displays Rewind, Play, and Fast-Forward buttons. The user clicks 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 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 clicks 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 game display data to the user (unless displaying the final state, in which case it does nothing).

**User clicks 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 game display data to the user (unless displaying the initial state, in which case it does nothing).

**Current game state is final game state:** The Game Display Module will not display the play button, preventing the user from playing the game any further.

USE CASE: PLAYBACK MODE PLAYING

**BASIC COURSE:**

The Game Display Module begins animating a new turn and sends the animation 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 and starts animating the next.

**ALTERNATE COURSES:**

**User presses the Pause button:** The Game Display Module starts the current game state, paused. 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 speed at which the game animations play. The Game Webpage depresses the Fast-Forward button and releases the Rewind button.

**User presses the Rewind button:** The Game Display Module runs the game animations in reverse, depresses the Rewind button, and begins to play the previous game state in reverse upon the end of the game state’s animations. The Fast-Forward button is released if depressed.

**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, and does not begin the next turn.