Use Case: Uploading a Bot

BASIC COURSE:

The user clicks on the “upload a bot” link. The user then selects their player bot from a file navigation menu to upload to the server. The user then enters a name that is associated with their player bot. The user then clicks the “submit” button. The client then uploads the bot to the server. The server then validates the bot. The server sends a message to the client stating “The bot was successfully uploaded.” The client then displays this message to the user.

ALTERNATE COURSES:

**Malicious bot detected:** The server rejects the player bot and displays an error message on the client stating, “The player bot you tried to upload was malicious and rejected.”

**Player bot is not a java file:** The server rejects the player bot and displays an error message on the client stating, “The player bot you tried to upload was rejected because it is not a java file.”

**File does not exist:** The client displays an error message stating, “The file you selected does not exist.”

USE CASE: USER INPUT IN TESTING MODE

**BASIC COURSE:**

The client begins a player-controlled character’s turn in a testing mode game. The client displays an input field for a command, a “submit” button, an undo button. The client then sends the user’s input to the server. The server validates the user’s input, evaluates it, and returns the resulting game data (game states, animation commands, and notifications). The server then sends this data back to the client. The client passes the data to the game display module, which displays it in graphical form. The system then begins the next turn.

**ALTERNATE COURSES:**

**Current turn is the first turn of the game:** The undo button is not displayed.

**Current turn is a previous turn which has been undone:** The redo button is displayed.

**User input was not valid:** The server will end the game and send a message to the client indicating that the input was invalid. The client will then display the notification “Error: Invalid Input” to the user.

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

Use Case: Client-side Bot Input in Testing Mode

BASIC COURSE:

After the user clicks the “Client Bot Input” button, the client will first prompt the user to modify their bot. The client shall then run the bot once, obtain the bot’s output, and then send the bot’s output to the server. The server will then validate the bot’s output, and determine if the output was a game winning move. The client will then prompt the game display module to end the current turn and halt any animations on screen. The client will then wait for a response from the server, and this process will repeat.

ALTERNATE COURSES:

**The user does not modify their bot:** The client will run the bot that was originally submitted to the Testing Arena. The client will make no further changes to the Basic Course scenario.

**The game has been already won:** The client will display the message “Game finished” to the user. The client will refuse any further input from the bot, and will return the user to the Testing Arena main menu.

**The bot input was not valid:** The client will display the error message “The bot input was not valid” to the user. The client will refuse any further input from the bot, and will return the user to the Testing Arena main menu.

**The bot takes too long to produce output:** The client will display the error message “The bot failed to quickly produce output” to the user. The client will refuse any further input from the bot, and will return the user to the Testing Arena main menu.

Use Case: Server-side Bot Input in Testing Mode

BASIC COURSE:

After the user clicks the “Server Bot Input” button, the client will prompt the user to proceed to the next turn. The client shall then run the bot once, after which the client will send the bot’s output to the server. The server will then validate the bot’s output, and determine if the output was a game winning move. The client will then prompt the game display module to end the current turn and halt any animations on screen. The client will then wait for a response from the server, and this process will repeat.

ALTERNATE COURSES:

**The game has been already won:** The client will display the message “Game finished” to the user. The client will refuse any further input from the bot, and will return the user to the Testing Arena main menu.

**The bot input was not valid:** The client will display the error message “The bot input was not valid” to the user. The client will refuse any further input from the bot, and will return the user to the Testing Arena main menu.

**The bot takes too long to produce output:** The client will display the error message “The bot failed to quickly produce output” to the user. The client will refuse any further input from the bot, and will return the user to the Testing Arena main menu.

Use Case: Challenge Mode

BASIC COURSE:

The client displays the menu of game modes and options to the user through the Game Display Module. The user selects Challenge mode to play. The client will display a list of challenges and bots for the user to select. The user will select the challenge and the player bot they wish to use. The client will send the selected bot and challenge to the server.

The server will run the challenge mode game with the selected bot and challenge by passing commands produced by the bots to the game evaluation module. The game evaluation module will produce a new game state and return it to the server. The server will send the game data (states, commands, and errors) to the client. The client will pass game states to the Game Display Module. The Game Display Module will create the display using these game states and display to the user using the Game Display Window. The client will pass a confirmation message to the server to alert the server that it can send another game state. This process will repeat until the game has been completed.

ALTERNATE COURSES:

**Bot input produces error:** The server will treat the command as “idle” and pass this command to the game evaluation engine.

**Turn limit exceeded:** The server will automatically terminate any games lasting a large preset number of terms (currently 500) as a precaution to an endless stalemate.

**Bot takes too long to produce output:** The server will terminate a bot that takes too long to produce a command and produce an error stating “Server-side bot did not produce command fast enough”.

Use Case: Replay Mode

BASIC COURSE:

The client displays the menu of game modes and options to the user through the Game Display Module. The user clicks on Replay Mode. The client displays a list of games to view. The user selects from a list of games that are able to be viewed and sends this to the client. The client sends a message to the server requesting the game data from the selected game. The server will send the client the game data to be viewed (states and commands). The client will accept this data and use the Game Display Module to create the display. The Game Display Module will send display to the user through the Game Display Window. The user will have the option to rewind, fast-forward, pause, and play the game with buttons provided.

ALTERNATE COURSES:

**Live game in replay mode:** The user will be unable to rewind, fast-forward, pause, and play if the game is in the process of being played.