Abdurrahman Mohammad

Eduardo Fragoso

Analysis Phase

Functional Specification

* Lets the user play tic tac toe with three different difficulty levels: easy, medium, hard
* Lets the user play tic tact toe with another player/user
* Lets the user load an existing game and finish it
* Lets the user save the current game

User Manual

1. Start the program. You will be greeted “Welcome to Tic Tac Toe!” and will be asked “Please select game mode”. You will be given options as such: easy, medium, hard, player vs. player. Click the button of the game mode of your choice.
2. You will then be asked to load an existing game. If you select yes, type in the name of the file you wish to load. The program will attempt to load your file. If it succeeds, you will be able to finish your saved game. If the program fails to load your game, a new one will be created, and the program will continue as if you have selected no.
3. You will then be asked if you would like to start first. If you select yes, your playing piece will be ‘X’ and the AI’s playing piece will be ‘O’. You will be the first to make a move on the board.
4. If the user has loaded an existing game, the program will check if the game is completed. If it is, the winner will be displayed. If not, the user will be able to complete the game by switching turns with the AI or another user (Player vs. Player) until somebody wins, the game ends in a draw, or if the user wishes to save and exit. If a new game has been created, the user and the AI or another user will exchange turns until somebody wins, the game ends in a draw, or if the user wish to save and exit.
5. If the user saves an exit, the user will be prompted to enter a filename. The system will then attempt to save the file. If it succeeds, it displays and success message and exits. If it fails, it displays an error message and exits.
6. If the game ends as usual with a win, loss, or draw, the user will be asked if he wishes to save the game. If the user selects yes, the user will be asked to enter a filename. The system will then attempt to save the file. If it succeeds, it displays and success message. If it fails, it displays an error message.
7. If the game ends as normal, without the save and exit button, the system will ask the user if he wishes to play again. If the user selects no, the system will exit and close. If the user selects yes, the process will be repeated from step 1.

Use Cases

Notes:

* The GUI is documented thoroughly. All options for button presses are accounted for.

**Use Case: Program start**

At the start of the program, the user can choose 1 of the 4 game modes: easy, medium, hard, player v. player. The player can also save and stop the program. The user is then asked if he wishes to load a file.

**Use Case: Close the window**

Anytime you close the window, the program will shut down.

**Use Case: Save and Stop**

The user will be asked to type in a filename. Then the program will attempt to save the file. If the save was successful, a success message is displayed. If the save failed, an error message is displayed. The system creates a file to store the game and then exits.

**Use Case: Easy**

An AI with easy difficulty is created. The user is then asked if he wishes to load a file.

**Use Case: Medium**

An AI with medium difficulty is created. The user is then asked if he wishes to load a file.

**Use Case: Hard**

An AI with hard difficulty is created. The user is then asked if he wishes to load a file.

**Use Case: Player vs. Player**

The user will be allowed to verse another user. No AI is created. The user is then asked if he wishes to load a file. He may select yes, no, or he may save and exit.

**Use Case: load a file**

If the user answers yes to save a file, he enters the name of the file. The system then tries to retrieve and decipher the file. If the system succeeds, the user is asked if he wishes to go first to which he may answer yes, no, or he may save and exit. If the system fails to load the file, refer to the use case “do not load a file.”

**Use Case: do not load a file**

A new tic tac toe game is created, and the user is asked if he wishes to go first to which he may respond yes, no, or he may save and exit.

**Use Case: Start first**

If the user wishes to start first, his playing piece is “X” and he will be the first person to make a move. Then the AI or the second user makes his move. The cycle then continues until the game ends with either player winning, the game ending in a draw, or the user presses save and continue. If the user ends the game with save and continue, refer the use case for “save and continue.” If the game ends with a win or a draw, the user will then have the option to save the game to which he may respond with yes, no, or save and stop. Refer to “save and stop” for that case.

**Use Case: Do no start first**

If the user does not wish to start first, his playing piece is “O” and he will be the second person to make a move – after AI or the second user. Then the (first) user makes his move. The cycle then continues until the game ends with either player winning, the game ending in a draw, or the user presses save and continue. If the user ends the game with save and continue, refer the use case for “save and continue.” If the game ends with a win or a draw, the user will then have the option to save the game to which he may respond with yes, no, or save and stop. Refer to “save and stop” for that case.

**Use Case: Save game after finishing**

The user is asked for a filename and the game is saved if the filename is correct. If the file saves or not, refer to “Do not save game after finishing” for the next steps.

**Use Case: Do not save game after finishing**

The user is then asked if he wishes to play again to which he may select yes, no, or save and stop.

**Use Case: Play again**

The whole program is started again. Refer to the first use case.

**Use case: Do not play again**

The system is then exited and the program stops.

Use Cases Diagram

A close up of a map

Description automatically generated