Fully Dressed Use Case

Group No# 9

Members:

Abdur Rahman Abul Hossain
Muhammad Hammad
Muhammad Uwais Jahmeerbacus
Zachary Cheema
Zoe Collins

Save current game

Primary Actor: User
Stakeholders and Interests:

- *User:* wants to save progress of the current game being played in order to return to the game later.
- *Players:* wants to save their current progress in the game to be able to return later.
- *Software Developers:* wants the game to start without any errors.
- *UI Designers:* wants the interface to feel interactive and user friendly.

Preconditions:

- Open Blokus Game use case.
- A game has been started.
- A game still has remaining moves

Postconditions:

- Positions of pieces placed on the board are saved.
- Current turn of a player is saved.
- The number of players is saved.
- The colours for every player is saved
- The AI difficulty is saved.
- All the available pieces are saved.

Main Success Scenario:

1. The system allows the user to save the current game.
2. User selects the option to save.
3. The system prompts the user to input a name for the save file [Alt 1: User selects cancel while in the save window].
4. The system obtains the time and commits that information to the save file [Alt 2: System is unable to obtain the time].
5. The system obtains the date and commits that information to the save file [Alt 3: System is unable to obtain the date].
6. The system obtains the number of players and commits that information to the save file.
7. The system obtains the number of human players and commits that information to the save file.
8. The system obtains the number of computer players and commits that information to the save file.
9. The system obtains the colour for each player and commits that information to the save file [Alt 4: System is unable to obtain the colour].

10. The system obtains if color blind mode is enabled and commits that information to the save file [Alt 5: System is unable to obtain if color blind mode is enabled].
11. The system obtains computer difficulty level and commits that information to the save file.
12. The system obtains positions of the current pieces placed on the board and commits that information to the save file.
13. The system obtains remaining pieces for each player and commits that information to the save file.
14. The system obtains which player's turn it is and commits that information to the saved file.
15. The system returns the user back to the game then the game continues.

Alternative Flows:

1. Alt 1: User selects cancel while in the save window.
    - The current game continues and the use case ends
2. Alt 2: System is unable to obtain the time.
    - System prompts the user to input the time.
3. Alt 3: System is unable to obtain the date.
    - System prompts the user to input the date.
4. Alt 4: System is unable to obtain the colour
    - System informs the user it was unable to save the current game.
    - Error log is created and saves the information for developers review.
5. Alt 5: System is unable to obtain if color blind mode is enabled
    - System informs the user it was unable to save the current game.
    - Error log is created and saves the information for developers review.

Exceptions:

- If at the moment when the user is prompted to input a name for the save file and nothing is entered then the system informs the user to input a valid name for the save file.

Special Requirements:

- The system is able to export information to a csv files to save piece positions on the board.

Open Issues:

- Is the user able to save a game by overwriting an existing save?