2 Fully Dressed Use Cases(Revised)

Revised Fully Dressed Use Case: Initiate a Game

Primary Actor:

The player

Stakeholders and Interests:

The human player

The AI players

Preconditions:

The user must be on the main menu screen

Postconditions:

On success the match will load, and the first player will be selected to take their turn

Main Success Scenario:

1. The user opens the game
2. The system shows user game selection screen
3. The system displays the options of Starting the game and existing the game
4. The user selects the difficulty level of the game.
5. The user inputs the number of players
6. The user inputs the game difficulty
7. The user inputs the score type
8. The user inputs the color of their pieces
9. The user inputs the colorblind settings if required (**Alternative flow: User decides to change the settings)**
10. The user clicks the start button (**Exception: Incorrect input**)
11. The system initializes the new game and displays the board of the game

Alternative Flow:

User decides to change the settings

(Alternative flow starts from the 9th step of the main success scenario)

1. The user decides to make changes to the settings before clicking start
2. The user changes the game settings input
3. Returns to step 10 in the main scenario

Exceptions:

Incorrect input

(*The exception starts from step 10 of the main scenario***)**

1. The user clicks the start button
2. The system has detected input that has not been selected
3. The system gives an error message to the user alerting them of what they need to enter
4. Returns to step 3 in the main scenario

Special Requirements:

Based on the colour-blind settings, change the colours that represent the players

Open Issues:

Will there possibility of networked players?

How is the order of player's turns decided?

Revised Fully Dressed Use Case: Take a Turn

Primary Actor:

The human or AI taking the current turn

Stakeholders and Interests:

The player or players

Preconditions:

The match must be started

Postconditions:

The next player's turn starts

Main Success Scenario:

1. The system asks the player to make a move or take a turn.
2. Turn starts with a user as the player
3. The user selects a piece **(Exception: No more pieces left)**
4. The user is given the option to rotate,flip and choose the placement of the piece on the board.
5. The user inputs their decision for the turn
6. The system checks the placement of the piece **(Alternative flow: Invalid placement of piece, make valid placement)**
7. The player's piece is placed on the board
8. The system tells the player their turn is over
9. The systems starts the next player's turn
10. The players keep on taking turns until one of the players is done placing all the pieces on the board
11. The system announces the player, who has finished placing all the pieces first or has the least amount of pieces left, as the winner.

Alternative Flow

Invalid placement of piece, make a valid placement

(The alternative flow starts from the 6th step of the main scenario)

1. The system checks the placement of the piece

2. If the placement is invalid, the system asks the player to rotate or flip or change the piece selected.
3. The steps from step 7 of Main success scenario is executed.

Exceptions:

No more pieces left:

(The exception starts from step 3 of the main scenario)

1. Final player in the game is not able to lay a piece
2. System counts the score for each player
3. System announces a winner
4. System gives option to replay or go to the main menu
5. This use case ends

Special Requirements

If the current player taking the turn is not human, have the program decide how the turn will be taken.

Open Issues

How will the AI decide to take its turn?