Use case description: Take a turn.

Primary Actor: Player

Stakeholders and Interests:

- Player: Wants to be a to take a turn by moving/rotating one of their pieces and placing it onto the board.
- Other Players(AI): Want to know what spaces are available to lay a piece, and to lay that piece according to the board layout and difficulty.

Preconditions:

- Player must select difficulty, board must be loaded.

Postconditions:

- The piece was placed on the board or the board is full and winner is determined.

Main Success Scenario:

1. The system tells the player it's their turn.
2. The player selects a piece to be placed on the board.
3. The system allows the player to rotate and flip the piece, then to select where the piece will be placed.
4. The player selects orientation and selects where the piece will be placed on the board.
5. The system validates the placement of the piece. [*Alt1: Invalid placement of the piece*]
6. The system places the piece on the board.
7. The system tells the player their turn is over and the player whose turn it is next. [*Alt2: No more players able to place pieces*]

Alternative Flows:

Alt1: Invalid placement of the piece

1. The system checks to see if there are any valid placements left for the piece in the players hand and the other pieces yet to be placed. [*Alt2: No moves remaining*]
2. The system informs the player that the placement is invalid, then allows the player to re-orientate and select a new placement.
3. Flow resumes at Main Success Scenario Step 5.

Alt2: No more players are able to place pieces

1. If there are no players left that are able to place any more pieces, the system ends the game.
2. Use Case Ends.

Exceptions:

- No more moves can be made by any player ending the game.

Special Requirements:

- Colours of pieces used must provide - or be able to provide - for the visually impaired (e.g. colour blindness).

Open Issues:

- Make sure that there is no way for turns to overlap.
- Figure out a way to select which player goes first randomly.