Case1: Place a Piece

Primary Actor: Player

Stakeholders and Key Interests:

- Player: Wants to place a piece on the game board.
- Developer: Wants to take feedback from the player to find what was difficult/can be improved and solve the bugs
- Mark Hatcher and TAs: Want to monitor the development of the game and grade it.

Preconditions:

- A game must be started and still in progress. There must also be a move left to make.

Success Guarantee(Postconditions):

- The system successfully places a piece on the board depending on the shape and size of the selected piece.

Main Success Scenario:

1. The player selects the shape they would like to place.
2. The system loads this piece.
3. The player selects where they want to place the piece .[Alt 1: Square is taken]
4. The system verifies that the move is valid. [Alt2: Move is invalid]
5. The system asks the player to confirm the move.[Alt3: Cancel move]
6. The user confirms the move.
7. The system places the piece on the board.
8. The system passes the turn to the next player. [Use case ends]

Alternative Flows:

Alt 1 : Square is taken

1. The system displays a message that the square is occupied.
2. The player places the shape on the board.
3. Flow resumes at step 4.

Alt 2: Move is invalid

1. The system displays a message that the move is illegal
2. The player places the shape on the board
3. Flow resumes at step 4

Alt 3: Cancel move

1. The player clicks the cancel button
2. The system exits the confirmation screen
3. Flow resumes at step 3.

Exceptions:

- If for some reason the connection is lost or the game fails, the use case ends.
- If the time runs up, the use case ends.

Special Requirements:

- Blocks can have numbers for colour blind players and can provide colours and sizes of text fonts used.
- Confirmation of move can have colour buttons to confirm/cancel move.

Open Issues:

- Is the player provided with enough hints on where a piece could be placed?
- What if the game crashes while on the confirmation screen?

Case2: Rotate/Flip a Piece

Primary Actor: Player

Stakeholder and Key Interests:

- Player: wants to rotate or flip a piece
- Developer: Wants to take feedback from the player to find what was difficult/can be improved and solve the bugs
- Mark Hatcher and TAs: Want to monitor the development of the game and grade it.

Preconditions:

- A game must be started and there must be a move left to make. A piece must also be selected.

Success Guarantee(Postconditions):

- A piece is successfully flipped or rotated and is ready to be placed on the board.

Main Success Scenario:

1. The player selects to the piece to be rotated[Alt1:Piece is a square]
2. The system loads the piece
3. The player selects the rotate-clockwise button
4. The system rotates the piece clockwise.
5. The player selects the flip button
6. The system flips the piece upside-down
7. The player selects the rotate counter-clockwise button
8. The system rotates the piece counter-clockwise
9. The player selects the flip piece
10. The system flips the piece right-side up

11. The system loads the piece and is ready to be placed[Use case ends]

Alternative Scenarios:

Alt1: Piece is a square

1. The system determines the piece is a square
2. The system loads the piece but does not rotate it
3. Flow resumes at step 11

Exceptions:

- If the connection is lost or the game fails, the use case ends.
- If the time runs up, the use case ends.

Special Requirements:

- The player should be able to flip and rotate the piece as many times as they wish.

Open Issues:

- Is it better to not flip a square or to flip it anyway?
- Will flipping/rotating a specific piece overlap anything in the GUI and cause problems?
- Will there be a delay once a piece is rotated or flipped to provide clear transition or will this slow down the pace of the game?