

# Revised Two Fully Dressed Use Case Description

## Initiating the game:

### Primary actor :

Player

### Stakeholders & Interest :

Developers: Wants the Blokus game to run efficiently.

Player: Wants to enjoy a bug free game.

### Preconditions :

User is able to run the game.

System requirements for the game are met.

### Postconditions :

User has begun the game.

Number of player/s is set.

Level of difficulty is selected.

### Main success scenario:

1. User launches the game
2. System initializes and gives the user the option to load a saved game or start a new one.
3. User selects to start a new game. [Alt1: User selects to load a saved game]
4. The system requires the system to select between normal play and colour blind play.
5. The user selects normal play.
6. System requires user to select a difficulty level.
7. User selects the difficulty level.
8. System requires the user to select the number of player/s with a minimum of 1 and a maximum of 4.
9. User chooses the number of players.
10. System gives the choice to start the game
11. User decides to start. [Alt2: The user decides not to play]
12. System initiates the game.

### Alternative Flows:

Alt1: The user selects to load a saved game

- The player will directly jump to its previously saved game. Use case ends.

Alt2: The user decides not to play

- The player will be given an opportunity to save the game or exit the game without saving. Use case ends.

### Exception

If at any time the system is unable to retrieve, record or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

### Special requirements:

- The system gives the option of loading a saved game

-Colours of text fonts used must change in order to accommodate the needs of colour blind people.

## Take a turn:

**Primary Actor:** Player

**Stakeholders & Interest :**

**Developers:** Wants the Blokus game to run efficiently and debug the game if any bugs are found.

**Player:** Wants to enjoy a bug free game.

**Precondition :**

The system validates if there is an available move.

The user wants to continue playing.

**Post condition:**

The turn ends and passes the turn to the next person.

The next player wants to continue playing.

**Main Success Scenario:**

1. The system indicates that it's the player's turn.
2. The user has the option to close the game or to continue playing.
3. The user continues playing and selects a block to place on the board. [Alt1: The user closes the game]
4. The system gives a hint where the block can be placed. [Alt2: The user disabled the Hints]
5. The user has an option to choose a different block.
6. The user has the option to flip or rotate the blocks.
7. The user selects a spot on the board to place the block.
8. The system determines if the placement is valid.
9. The system puts the block in the desired valid place.[Alt3- The placement is not valid]
10. The system removes the block from the block bank.
11. The system ends the current turn.
12. The system passes the turn to the next player.

**Alternative Flows-**

Alt1: The user decides to close the game. The system gives the option to save or close.

Alt2: The user disabled the hints. They system doesn't provide any hint.

Alt3: The spot the user selected is invalid. The system has to inform the player to choose a valid location shows possible placements.

**Exception-**

If at any time the system is unable to retrieve, record or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements-**

1. Gives the option to save.