

Use case descriptions for "Initiate a game" and "Take a turn":

"Initiate a game":

Primary actor:

-User

Stakeholders and interests:

-User: Wants to initiate a game of Blokus with a desired number of human players.

-Development team: Wants the game to initiate properly to allow the user(s) to play Blokus without bugs or errors.

Preconditions:

-The user has successfully ran the Blokus game program.

Postconditions:

-The user is able to play a game of Blokus without errors or bugs.

Main success scenario:

1. The system provides the user with the option of initiating a 2, 3 or 4 player game of Blockus (i.e. total number of players to be occupied by humans or computers) or to resume a previously saved game. [Alt 1: user chooses to resume a game]
2. The user then selects an option.
3. The system asks for the number of human players that will be playing this game.
4. The user enters a valid number of human players. [Alt 2: Invalid number of human players]
5. The system fills any remaining player slots with computer players and allows each player a choice of piece color (possibly 2 colors depending on number of players).
6. The system provides a selection of difficulty options for computer players if any player slots were filled by computer players.

7. The user selects a difficulty option for the computer players.
8. The system provides the options of basic scoring and advanced scoring to the user.
9. The user selects a scoring option.
10. The system sets up and initiates a game of Blokus configured to the settings chosen by the user and randomly selects 1 of the players to go first.

Alternative flows:

Alt1: User chooses to resume a game

1. The system provides the user with the available saved games.
  2. The user chooses a saved game to load.
  3. The system sets up a game of Blokus to match the state of the saved game chosen by the user.
- The use case then ends.

Alt2: Invalid number of human players

1. The system prompts the user to enter a valid number of human players according to the total number of players option selected by the user previously.
2. Flow resumes at Main success scenario step 3.

Exceptions:

-N/A currently

Special requirements:

-Must have color options to accommodate visual impairments such as color blindness.

Open issues:

-Should the system provide options for the number of human players that are dependent on the number of players chosen by the user or let the user enter the number and then verify that it is valid.

-----

"Take a Turn":

Primary actor: player (human or possibly computer)

Stakeholders and interests:

- Player: Wants to be able to take their turn without difficulty or bugs. Wants their moves and the moves of other players to abide by the rules of the game.
- Development team: Wants players to have a smooth experience playing Blokus. Wants to ensure that all moves made by human and/or computer players are legal and are exactly how the player intended.
- Wants the game board to update properly and in a manner that allows other players to see what move was made.

Preconditions:

- The game board is properly set up (or is updated correctly from the last turn made).

Postconditions:

- The move made is ensured to be legal and other players are aware of the move that has been made during the turn.
- The game board is updated and ready for the next turn.

Main success scenario:

1. The system indicates whose turn it is.
2. The system determines if the player has any possible moves that can be made. [Alt1: No possible moves]
3. The player who has the current turn can then select a piece to place on the board and attempt to make their move. The player may also choose to get a hint from the system. [Alt2: The player requests a hint]
4. The system validates the move made by the player and if the move is valid the player's turn is over. Finally the player's available pieces are updated. (use case ends). [Alt3: The player's move fails validation]

Alternative flows:

Alt1: No possible moves

1. The system informs the player that they have no possible moves and their turn is skipped.

The use case then ends.

Alt2: The player requests a hint

1. The system suggests a possible legal move that the player can make.
2. Flow resumes at Main success scenario step 3.

Alt3: The player's move fails validation

1. The system informs the player that their move was illegal.
2. Flow resumes at Main success scenario step 3.

Exceptions:

-N/A currently

Special requirements:

-N/A currently

Open issues:

- Should a hint suggest the best move, just any move, etc. to a player.
- Should the system inform a player why their move was invalid.