

Use Case Description: Initiate a Game

Primary Actor: Player

Stakeholders and Interests:

- Player: play the game of Blokus
- Computer Player: play the game against the human player(s).
- Developers: maintains the game and make changes.

Pre-condition:

- User have the appropriate system required to execute the game file.

Success Guarantee:

- The board is displayed.

Main Success Scenario:

1. The user selects to start a new game.
2. The system provides the user with settings to select including difficulty, number of players and colorblind modes.
3. The user selects difficulty, number of players and a colorblind mode.
4. The system creates computer players with difficulty chosen.
5. The system sets the color of players as chosen.
6. The system displays the grid.
7. The system allows the user to take the first turn.

Alternative flows:

Alt 1: User selects 'Resume Game':

1. The user selects to resume a saved game.
2. The system retrieves the game settings and positions of each block and resumes the game from last progress

Alt 2: User chooses to view rules

1. The user selects to view rules from the top menu bar.
2. The system displays the rules in a new window.

Exception:

- The user may try to resume the game when no game has been saved.
- The user exits the game mid-play without saving.

Open Issues:

- Resuming game from a saved file and generating the same state as before can be difficult.

Use Case Description: Take a Turn

Primary Actor: Player

Stakeholders and Interests:

- Player: play the game of Blokus
- Computer Player: play the game against the human player(s).
- Developers: maintains the game and make changes.

Preconditions:

- The user has initiated a game.
- It is the player's turn.

Success Guarantee:

- The player successfully placed a piece on the board.

Main Success Scenario:

1. The system provides the user with a list of piece to choose from.
2. The user selects a piece.
3. The system prompts with the option to rotate or flip the piece.
4. The user chooses to not rotate the piece.
5. The system allows the user to place the piece on the board.
6. The user places the piece on the board.
7. The system checks if the move is valid and takes the placement into account.

Alternate flows:

1. Player chooses to rotate the piece
 - a The user selects to rotate the piece.
 - b The system rotates the tile selected.
2. The move is not legal:
 - a The system detects the move is not valid.
 - b The system informs that to the user and allows the user to place tile again.

Exception:

- If the user tries to play a move when it is not his turn, the system will inform the user.

Open Issues:

- Determining a legal move played efficiently might be difficult.