Calculate scores:

Primary Actor: Player

Stakeholders and Interests:

-Player: wants to find out who has won the game, wants to see placement of other players (who came 2nd and so on).

-Development team: wants score calculation to be done according to the option chosen by the player (basic or advanced), wants scores to be correct and adhere to scoring rules of Blokus.

Preconditions:

-The player has selected either basic or advanced scoring when initiating the game.

-Neither player is able to place any more pieces.

Postconditions:

-Player is aware of who won the game and the placement of the other players along with the scores of each player.

Main Success Scenario:

1. The system informs the player that the game is over because there are no more possible moves and waits for the player to proceed to the score calculation.

2. The player confirms that they would like to proceed to the calculation of scores.

3. If basic scoring was chosen during setup of the game, the system calculates the number of squares in the remaining pieces of each player. The player with the lowest number of squares is declared the winner with the second lowest being declared 2nd place and so on. The system then waits for the player to confirm that they'd like to return to the main menu [Alt1: Advanced scoring was chosen]

4. The player confirms that they would like to return to the main menu. [Use Case Ends]

Alternative Flows:

Alt1: Advanced scoring was chosen

1. The system counts the number of squares in the remaining pieces of each player where 1 square = -1 point. If a player has placed all of their pieces, they earn +15 points and +5 bonus points if the last piece they placed was the smallest (one square) piece.

2. The system then displays the score of each player and their ranking (1st 2nd, etc.) and waits for the player to confirm that they'd like to return to the main menu.

3. Flow resumes at Main Success Scenario Step 4.

Exceptions:

-N/A

Special Requirements:

-N/A

Open Issues:

-Should score calculations happen without informing the user or is it better that the user is told exactly what is happening.

Load a game:

Primary actor: User

Stakeholders and Interests:

-User: wants to be able load a game and resume where they left off.

-Development Team: wants the system to resume a previously saved game at the exact point where the game was saved.

Preconditions:

-The user has previously saved a game of Blokus.

Postconditions:

-The user is able to continue the loaded game of Blokus right where they left off.

Main Success Scenario:

1. The user selects the option to load a game.

2. The system retrieves and displays saved games that can be loaded. [Alt1: No saved games available]

3. The user selects one of the saved games to load.

4. The system loads the saved game thus allowing the user to continue the game exactly where it left off. [Use Case Ends]

Alternative Flows:

Alt1: No saved games available

1.) The system informs the user that there are no saved games available to load. The use case ends.

Exceptions:

N/A

Special Requirements:

N/A

Open Issues:

-Should saved games be ordered in a particular way (for example: by date, alphabetical order, etc.)