

NEW Use Case #2 - Save Game

Primary Actor: Player

Stakeholders and Interests:

Player: Wants to save the current state of their game.

Preconditions:

Assuming one human player.

A game has been started, but not completed.

At least a single turn has been made.

Success Guarantee:

Any human player can save the current instance of the game.

Main Success Scenario:

1. During gameplay, the user selects the "Save Game" button in the top right of the UI.
2. The system then saves the current instance of the game by writing to a text file. This includes:
 - a. The current state of the board, including all of the currently played pieces.
 - b. The current state of the score among players.
 - c. The current state of each player's Blokus piece bank.
 - d. The current settings that are applied to the player's game.
3. Upon save completion, the UI will then confirm that the game has been saved successfully by displaying a brief confirmation. Ex. "Save Successful".

Alternative Flows:

1. **Overwriting A Save** - Given the case that a game has previously been saved and a text file already exists with game data, the UI will prompt the player if they would like to overwrite the current save prior to executing their save.
2. **Error During Save** - Given the case that an error prevented the game from successfully saving, the UI will inform the user with a brief message such as "Save Failed".

Exceptions:

- The user at any time terminates the program.
- System Errors.