Case: Initiate a Game

Primary Actor: Player
Stakeholders and Interests:

- Player: wants to initiate a game and wants to play with player/computer.
- Developer: wants to take feedback from player/user and what they found annoying/difficult to find and solve the bugs.

- Mark Hatcher and TAs: wants to monitor the development of game and grade it.

Preconditions:

- Player must click the play button to enter start menu.

Success Guarantee (Postconditions):

- System successfully lets player enter the game depending on what settings the player chose.

Main Success Scenario:

1. The player clicks on play button.
2. The system gives the player the opportunity to set grid sizing, number of players, and CPU(s).
3. The player sets the grid size. (options like 20x20, 16x16, 24x24)
4. The system provides an option to select the number of players.
5. The player selects the number of players. (options include 1,2,3,4)
6. The system provides the player the opportunity to select the number of CPU(s) to select depending on the number of players he selected.
7. The player sets the number of CPU(s) [Alt1: Player sets no CPU i.e. more than one player].
8. The system asks the player whether to set a time limit or not.
9. The player then selects a time limit. (options are yes or no)
10. The system asks the player to set up time limit. [Alt2: Player sets no time limit].
11. The player then sets up the time limit.
12. The system provides player the opportunity to select difficulty of the CPU(s).
13. The player set the CPU(s) difficulty. (options are easy, medium, and hard)
14. The system asks the player to select the block color of player(s) or gives the opportunity to randomize it.
15. The player sets the block color for each player and starts the game. (options are red, green, blue, yellow) [Use case ends].
16. The system asks the player to set a time limit, for how long, and difficulty for the AI.
17. The system then asks the player to provide block color for each player [Alt3: Player chooses to randomize colors between all player(s)/CPU(s)].
18. The player sets the block color for each player and starts the game. [Use case Ends].

Alternative Flows:
Alt1: Player sets no CPU i.e. more than one player

1. The system disables the difficulty option for CPU if there are 4 players.
2. Flow resumes at Step 8.

Alt2: Player sets no time limit.

1. The system disables the minutes option and sets no time limit for the game session.
2. Flow resumes at Step 12.

Alt3: Player chooses to randomize colors between all player(s)/CPU(s).

1. The system randomly assigns colors to all players. [Use case Ends]

Exceptions:

- The game will not save the current session setting if the process is closed in between.

Special Requirements:

- Blocks can have numbers for color blind players and can provide colors and sizes of text fonts used.

Open Issues:

- Is the player provided with enough hints to start the game?

- How the game state is going to be loaded?

- If there are no saved game sessions will the load fail?

Case: Take a turn

Primary Actor: Player
Secondary Actor: Computer
Stakeholders and Interests:

- Player: wants to take a turn with a select shape.
- Developer: wants to take feedback from player/user and what they found annoying/difficult to find, if logic is working properly, and solve the bugs.

- Mark Hatcher and TAs: wants to monitor the development of game and grade it.

Preconditions:

- Player must setup game settings.

Success Guarantee (Postconditions):

- System successfully informs player that time limit is over or there no possible moves and surrenders the player and does the same for computer. Finally, the system declares who the winner is and gives an opportunity to play again.

Main Success Scenario:

1. The system sets up the game board, time limit, score, game entities such as player, and computer(s) and gives player the first turn based on the initiate game settings.
2. The system provides a list of shapes which can be selected, flipped, and rotated.
3. The player selects the desired shape and places it on the board. [Alt1: Player flips/rotates the shape]
4. The system verifies that the player's move is valid.
5. The system allows the placement of the piece on board and stores the information.
6. The computer player then takes a legal strategic turn based on player's turn.
7. When the time limit is over, the system decides who has fewer total blocks. [Alt 2: No space to place the left shape for player]
8. The system gives the player an opportunity to play again and declares the winner [Use Case Ends].

Alternative Flows:
Alt1: Player flips/rotates the shape.

1. The Player flips/rotates the selected shape.

2. The Player places the selected shape on the game board.

3. Flow resumes at step 4

Alt 2: No space to place the left shape for player].

1. The system automatically makes the player surrender and informs the player that he has no possible moves left.
2. The system lets computer(s) take their turns and when the computers have no possible move left, the system surrenders computer(s).
3. Flow resumes at Step 8.

Exceptions:

- If for some reason, any of the functionality fails during the gameplay and the game crashes then the player is provided an option to provide error logs to the developer.

Special Requirements:

- Blocks can have numbers for color blind players and can provide colors and sizes of text fonts used.

Open Issues:

- Is the player provided with enough hints to play the game or take a turn?
- Does the system ask player to save the game state before clicking on “x” button?
- If the game unexpectedly crashed is the game state saved?
- What if the player surrenders on the first turn?