

Use Case Descriptions

1. Set up a game

- Primary Actor(s)

Player(s) : User of the system who will open the system.

- Stakeholders and Interests

Player(s) : Users will open the game and select from the menu for desired playing settings. User will choose the number of human and computer players. User will select the difficulty of the game. User will select the colors of their pieces. User can see a list of game rules. User can load a saved game.

Members : Will validate group members" code to open the game.

Marker : Will satisfy marker's requirement for this process

- Postconditions

- Game is loaded or initialized.
- A previous game is loaded (user selects saved games).
- A new game is initialized. (User selects new game).
- The type of player and their position is set.
- Difficulty for each opponent is set.
- Turn order is generated.
- Starting position of each player is initialized.
- Board is displayed

- Main Success Scenario

- 1) The user initializes the game system.
- 2) The system provides the user opportunity to select a new game, previous game or exit the use case.
- 3) The user selects a new game.
- 4) The system provides the opportunity to select from two player or four player settings.
- 5) The user selects two players setting.
- 6) The system provides the opportunity to select player as human or AI.
- 7) The user selects players as human or human and AI.
- 8) The system provides the opportunity to select the difficulty settings for opponent.
- 9) The user selects the difficulty as desired.
- 10) The system provides the opportunity to select which player will take turn first.
- 11) The user selects player1 to take their turn first.
- 12) The system set up a game board window with two players and generate turn order starting from player1.

- Alternative Flows

Alt 1: *User selects previous game.*

- 1) After first two steps above, the user selects previous game.
- 2) The system loads the previous game window with saved action events.

Alt 2: *User selects four player settings.*

- 1) After first three steps, the user selects four player settings.

- 2) The system provides the opportunity to select players as human or AI.
- 3) The user selects players as all humans or humans and AIs.
- 4) The system provides the opportunity to select the difficulty settings for opponent.
- 5) The user selects the difficulty as desired.
- 6) The system provides the opportunity to select which player will take turn first.
- 7) The user selects player1 to take their turn first.
- 8) The system set up a game board window with four players and generate turn order starting from player1.

Alt 3: *User exits use case.*

- 1) System provides the option to close the game at any given user steps.
- 2) User selects to close the game.
- 3) System presents the option for users to confirm closing the game.
- 4) User confirms.
- 5) System close the game window and exits the program.

- Exceptions

- The system does not open or initialize.
- The system can not load the previous saved game.
- The system can not set up the board as per instruction.

- Special Requirements

Colors and sizes of the text fonts used must provide for the visually impaired. The pieces on the board must be distinguished by shape or color to separate from each player so that, everyone can understand the position of their pieces.

- Open Issues

If the user tries to open a previous game which was not recorded to the system then that function will not work or will show an error. The user can not select all players as AI; One human player must be selected.

2. Take A Turn

- Primary Actor(s)

Player(s) : User (human or non-human) moves their pawn to a particular direction.

- Stakeholders and Interests

Player(s) : User will recognize when their turns comes up. They
Will be able to select what they can do with their turn
From the available options at the given point of the
Game. These options will include movement of a piece
to the available places, placement of a wall according
to the game rules. Users will be able to see the hints
about their legal action that they can take in their turn.
User will also be able to exit the game at anytime
without affecting any other player's gameplay.

Members : Will validate group members" code to pawn
movement.

Marker : Will satisfy marker's requirement for this process.

- Preconditions

A game board is set up with required number of players. The user
must select the use case "take a turn" to move their pawn over the
use case "set up a wall". The user must wait for their turn to perform
this use case.

- Postconditions

Player's pawn will move to the desired direction according to the
game rules.

- Main Success Scenario

- 1) The system identifies the next players turn and provide the opportunity to select from the use cases “take a turn” or “set up a wall”.
- 2) The user selects “take a turn”
- 3) The system provide the opportunity to move the pawn to the right, left, forward, backward one block if the block is empty or select use case “jump over the player” if opponent is in the forward space or select use case “move in diagonal direction” if the opponent is in the forward space and there is a wall behind the opponent.
- 4) The user selects move forward.
- 5) The system moves the pawn forward one block.

- Alternative Flows

Alt 1: *User selects jump over the player*

- 1) After first three steps, the user selects the use case “jump over the player”
- 2) The system moves the pawn to second block from it’s initial block jumping over opponent’s pawn.

Alt 2: *User selects move in diagonal direction*

- 1) After first three steps, the user selects the use case “move in diagonal direction”
- 2) The system then provides the opportunity to select in which side of the opponent pawn the user wants to move given that the block is empty or reachable.
- 3) The user selects a side.

- 4) The system moves the pawn diagonally.

Alt 3: User exits use case.

- 1) System provides the option to close the game at any given user steps.
- 2) User selects to close the game.
- 3) System presents the option for users to confirm closing the game.
- 4) User confirms.
- 5) System close the game window and exits the program.

- Exceptions

- The pawn moves in any direction even the block is unreachable because of the walls.
- The pawn can not jump over opponent's pawn.
- The system prompts for the use case take a turn even if a player's pawn reaches to its winning block.

- Special Requirements

The system will show the moving direction options if the block adjacent to the pawn is empty and there is no wall in between. The system will prompt a message showing the pawn can not move to that block if the player still tries to move there. Each player should wait for their turn to come.

- Open Issues

This use case's functionality depends on other use cases too. The proper design of that has not yet been constructed. Also the full functionality of this use case depends on the limitations of the source code. s