

# Use Case Descriptions (Connect4)

## Use Case: Start New Game

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To initiate a new game, resetting the game board and allowing players to begin a fresh round.

**Preconditions:** The application is active and ready for a new game.

**Trigger:** The player selects the option to start a new game.

**Scenario:**

1. The player chooses to start a new game.
2. The system clears the game board and prepares it for new moves.
3. The initial state of the game board is displayed.

**Postconditions:** The game board is reset and ready for the first move.

**Exceptions:** None.

**Priority:** High – Essential for game functionality.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** At the beginning of each new game.

**Channel to actor:** Visual confirmation of a new game board.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** N/A

## Use Case: Display Game Board

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To provide a visual display of the current game board, enabling players to understand the game state.

**Preconditions:** The game has started, and at least one move has been made.

**Trigger:** The game begins or a move is completed.

**Scenario:**

1. The system generates and displays a visual representation of the board.
2. Column numbers and current piece placements are shown.
3. The system updates the display after each move.

**Postconditions:** The board is continuously visible, reflecting the latest moves.

**Exceptions:** Display errors that prevent the board from being shown accurately.

**Priority:** High – Vital for game clarity.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** Continuously, after each move and at game start.

**Channel to actor:** Visual display of the board.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** N/A

## Use Case: Player Makes a Move

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To drop a piece in a chosen column, advancing the game.

**Preconditions:** The game is in progress, and it is the player's turn.

**Trigger:** The player selects a column for their move.

**Scenario:**

1. The player selects a column to place their piece.
2. The system validates the move for legality.
3. If valid, the piece is added to the board in the selected column.

**Postconditions:** The move is recorded, and the board reflects the new state.

**Exceptions:** If the column is full, prompt the player to select another column.

**Priority:** High – Core game mechanic.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** Repeatedly, throughout the game.

**Channel to actor:** Confirmation of the move.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** None

## Use Case: Validate Move

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To ensure the move chosen is valid and can be executed on the board.

**Preconditions:** The player has selected a column for their move.

**Trigger:** The player attempts a move.

**Scenario:**

1. The player's move triggers a check if the selected column has space for a new piece.
2. If the move is valid, the system allows the piece to be placed.
3. If not, the system prompts the player to select another column.

**Postconditions:** Only valid moves are executed, preventing errors.

**Exceptions:** Full columns prevent the move; the player is prompted to try again.

**Priority:** High – Ensures fair gameplay.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** Every time a player makes a move.

**Channel to actor:** Visual prompt if the move is invalid.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** None.

## Use Case: Check Win

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To determine if a winning condition has been met after making a move.

**Preconditions:** A valid move has been made.

**Trigger:** Completion of the player's move.

**Scenario:**

1. The player's move triggers an analysis of the board for four consecutive pieces by the same player.
2. If a winning condition is met, it triggers the end game.
3. If no winning condition is met, the game continues.

**Postconditions:** The game state is updated based on the win check.

**Exceptions:** None.

**Priority:** High – Critical for game outcome.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** After every move.

**Channel to actor:** N/A (system-initiated).

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** None.

## Use Case: End Game (Win or Draw)

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To conclude the game if a player has won or if the board is full, resulting in a draw.

**Preconditions:** A win condition is met, or the board is full.

**Trigger:** A win is detected or no moves are left.

**Scenario:**

1. The player's move or full board triggers the end game.
2. The system halts further moves.
3. It displays the game result to the players.

**Postconditions:** The game concludes, with results visible to players.

**Exceptions:** None.

**Priority:** High – Essential for completing each game round.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** Once per game.

**Channel to actor:** Game result display.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** None.

## Use Case: Display Game Result

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To show the outcome of the game, indicating a winner or a draw.

**Preconditions:** The game has ended.

**Trigger:** The game reaches a conclusion (win or draw).

**Scenario:**

1. The system shows the final game result.
2. It indicates the winning player or a draw.
3. Optionally, players are given the option to start a new game.

**Postconditions:** Players are informed of the game's outcome.

**Exceptions:** None.

**Priority:** Medium – Important for player feedback.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** Once per game.

**Channel to actor:** Visual display of the result.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** None.

## Use Case: Switch Player Turns

**Iteration:** 1

**Primary Actor:** Player

**Goal in context:** To alternate between players after each valid move, ensuring fair gameplay.

**Preconditions:** A player has successfully completed a valid move.

**Trigger:** A valid move is executed by the current player.

**Scenario:**

1. The current player completes a valid move.
2. The system identifies the next player in turn.
3. The system updates the active player indicator, signalling whose turn it is.
4. The new active player is allowed to make their move.

**Postconditions:** The turn alternates to the next player, allowing them to proceed with their move.

**Exceptions:** If a turn switch error occurs, the system reverts to the previous player for correction.

**Priority:** High – Ensures balanced and structured gameplay.

**When available:** Within 1 sprint (first iteration).

**Frequency of use:** Every turn, throughout the game.

**Channel to actor:** Visual or indicator prompt showing the current player.

**Secondary actors:** N/A

**Channel to secondary actors:** N/A

**Open issues:** None.