### **Use Case: Choose a Game**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to select a game to play from the available options.

**Preconditions**: The user must be on the main menu screen.

**Trigger**: The user clicks the "Choose a Game" button.

**Scenario**:

1. The user clicks the "Choose a Game" button.
2. The system displays a screen with options for available games (e.g., Tic-Tac-Toe, Connect 4).

**Post Conditions**: The user is presented with a game selection screen.

**Exceptions**:

* The application fails to load the game selection screen due to an error.
* **Priority**: Medium priority, as it provides access to game functionality.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Multiple times per session if the user chooses to explore different games.

**Channel to Actor**: User clicks the "Choose a Game" button on the screen.

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Show Game Menu**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to navigate to a screen where they can choose a game to play.

**Preconditions**: The application must be running, and the user must be on the main menu screen.

**Trigger**: The user clicks the "Choose a Game" button.

**Scenario**:

1. The user clicks the "Choose a Game" button.
2. The application displays the game selection menu with available game options such as Tic-Tac-Toe and Connect 4.
3. The user can click on any game button to proceed.

**Post Conditions**: The game selection screen is displayed, allowing the user to choose a game.

**Exceptions**:

* The screen fails to load due to an error in the system.

**Priority**: Medium priority, as it facilitates access to game-playing features.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Once or multiple times per session, depending on user preference.

**Channel to Actor**: User clicks the button on the main menu screen.

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Show Settings Menu**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Provides the user with an option to access application settings.

**Preconditions**: The user must be on the main menu screen.

**Trigger**: The user clicks the "Settings" button.

**Scenario**:

1. The user clicks the "Settings" button.
2. The application displays the settings screen.
3. The user can adjust settings as needed (placeholder for future enhancements).

**Post Conditions**: The settings screen is shown, allowing the user to review or adjust settings.

**Exceptions**:

* The settings screen fails to load due to an internal error.

**Priority**: Low priority, as it is an auxiliary feature.

**When Available**: Within its 1st iteration.

**Frequency of Use**: As needed by the user for configuration purposes.

**Channel to Actor**: User interacts with the button on the main menu screen.

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Show Help Menu**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to access help or information about using the application.

**Preconditions**: The user must be on the main menu screen.

**Trigger**: The user clicks the "Help" button.

**Scenario**:

1. The user clicks the "Help" button.
2. The help screen is displayed, providing information or instructions.
3. The user can read the information and click "Back to Main Menu" to return.

**Post Conditions**: The user can access and review help content.

**Exceptions**:

* The help screen fails to load due to an error.

**Priority**: Medium priority, as it aids users who need assistance.

**When Available**: Within its 1st iteration.

**Frequency of Use**: As needed by the user for reference or assistance.

**Channel to Actor**: User clicks the button on the main menu screen.

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Handle Move in Tic-Tac-Toe Game**

**Iteration**: 2

**Primary Actor**: User

**Goal in Context**: Allows the user to make a move in the Tic-Tac-Toe game.

**Preconditions**: The user must be on the Tic-Tac-Toe game screen.

**Trigger**: The user clicks a cell on the game board.

**Scenario**:

1. The user clicks an available cell on the game board.
2. The system marks the cell with the current player's symbol (X or O).
3. The system checks if the move results in a win or draw.
4. If a win is detected, the system displays a win alert and resets the board.
5. If a draw is detected, the system displays a draw message and resets the board.
6. If no win or draw is detected, the turn switches to the other player.

**Post Conditions**: The game state updates with the user's move, and the game progresses.

**Exceptions**:

* The cell is already marked and cannot be selected.

**Priority**: High priority, as it is essential for game play.

**When Available**: Within its 2nd iteration.

**Frequency of Use**: Multiple times per game session.

**Channel to Actor**: User clicks a button on the game board.

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Show Win Alert for Tic Tac Toe**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Notifies the user when a player wins the Tic-Tac-Toe game.

**Preconditions**: The user must be playing a game, and a win condition must be detected.

**Trigger**: The system detects a winning combination.

**Scenario**:

1. The system checks for a winning condition after a move.
2. If a win is detected, an alert is shown with the winning player's name.
3. The user is given options to either play again or exit.
4. If the user chooses to play again, the board is reset.
5. If the user chooses to exit, the application closes.

**Post Conditions**: The user is informed of the game result and can choose to continue or exit.

**Exceptions**:

* The alert fails to display due to an internal error.

**Priority**: High priority, as it is crucial for game result notification.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Once per game when a win condition is met.

**Channel to Actor**: System displays an alert on the screen.

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Send Message in Chat**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to send a message in the chat area of the Tic-Tac-Toe game.

**Preconditions**: The user must be on the Tic-Tac-Toe game screen with the chat feature visible.

**Trigger**: The user types a message and presses "Enter" or clicks the "Send" button.

**Scenario**:

1. The user types a message in the chat input field.
2. The user presses "Enter" or clicks the "Send" button.
3. The system appends the message to the chat area.
4. The input field is cleared for the next message.

**Post Conditions**: The message is displayed in the chat area.

**Exceptions**:

* The input field is empty, and no message is sent.

**Priority**: Low priority, as it is an additional feature for interaction.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Multiple times per game session as desired by the user.

**Channel to Actor**: User types and clicks the "Send" button or presses "Enter".

**Secondary Actors**: N/A

**Channel to Secondary Actors**: N/A

**Open Issues**: N/A

### **7. Use Case: Exit Application**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Close the application.

**Preconditions**: User is on any screen.

**Trigger**: User clicks the "Exit" button.

**Scenario**:

1. User clicks the button.
2. Application closes.

**Post Conditions**: Application is terminated.

**Exceptions**:

* Application fails to close.

**Priority**: High

**When Available**: 1st iteration

**Frequency of Use**: Once per session.

**Channel to Actor**: User clicks the button.

**Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Handle Move in Connect4 Game**

**Iteration**: 2

**Primary Actor**: User

**Goal in Context**: Allows the user to make a move in the Connect4 game.

**Preconditions**: The user must be on the Connect4 game screen.

**Trigger**: The user clicks a column on the game board.

**Scenario**:

1. The user clicks an available column on the game board.
2. The system drops the current player's disc into the lowest available row in that column.
3. The system checks if the move results in a win (four consecutive discs of the same color) or draw.
4. If a win is detected, the system displays a win alert and resets the board for a new game.
5. If a draw is detected, the system displays a draw message and resets the board.
6. If no win or draw is detected, the turn switches to the other player.

**Post Conditions**: The game state updates with the user's move, and the game progresses.

**Exceptions**:

* The column is full and cannot accept more discs.

**Priority**: High priority, as it is essential for game play.

**When Available**: Within its 2nd iteration.

**Frequency of Use**: Multiple times per game session.

**Channel to Actor**: User clicks a column on the game board.

**Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Handle Move in Checkers Game**

**Iteration**: 2

**Primary Actor**: User

**Goal in Context**: Allows the user to make a move in the Checkers game.

**Preconditions**: The user must be on the Checkers game screen.

**Trigger**: The user selects a piece and a destination cell on the board.

**Scenario**:

1. The user selects an available piece to move.
2. The system highlights valid move destinations for the selected piece.
3. The user clicks on a valid destination cell.
4. The system moves the piece to the selected cell.
5. The system checks if the move results in a capture or win condition.
6. If a capture is made, the system removes the opponent's piece and checks if another capture is possible for the same piece.
7. If a win condition is met (all opponent's pieces captured or blocked), the system displays a win alert.
8. The turn switches to the other player if no further moves are possible for the current turn.

**Post Conditions**: The board updates with the move, and the game continues.

**Exceptions**:

* The selected move is invalid (e.g., not following game rules).
* The user attempts to move an opponent's piece.

**Priority**: High priority, as it is essential for game play.

**When Available**: Within its 2nd iteration.

**Frequency of Use**: Multiple times per game session.

**Channel to Actor**: User clicks a piece and a destination on the game board.

**Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Challenge Other Player by Searching Their Profile**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to search for another player’s profile and send them a game challenge.

**Preconditions**: The user must be logged in and on the main menu, profile screen, or another screen with access to the search functionality.

**Trigger**: The user enters the name or ID of the other player in the search bar and selects their profile.

**Scenario**:

1. The user enters the name or ID of another player in the search bar and initiates the search.
2. The system retrieves and displays the search results.
3. The user selects the desired player’s profile from the search results.
4. The system displays the selected player’s profile, including their game statistics and options for interaction.
5. The user clicks the "Challenge" button on the profile page.
6. The system sends a challenge notification to the selected player.
7. The selected player receives the challenge and can accept or decline it.

**Post Conditions**: The challenge is sent, and the system waits for the response from the challenged player.

**Exceptions**:

* The search returns no results due to incorrect input or network issues.
* The selected player is not available for challenges (e.g., offline or has restricted their profile).
* The challenge fails to send due to a server or network error.

**Priority**: High priority, as it promotes user interaction and gameplay initiation.

**When Available**: Within its 1st iteration.

**Frequency of Use**: As needed by the user when searching for and challenging opponents.

**Channel to Actor**: User interacts with the search bar and selects the profile, then clicks the "Challenge" button.

**Secondary Actors**: The challenged player.

**Open Issues**: N/A

### **Use Case: Access Game Library**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to browse the library of available games to select and play.

**Preconditions**: The user must be logged in and on the main menu or another accessible screen.

**Trigger**: The user selects the "Game Library" option from the main menu or relevant interface.

**Scenario**:

1. The user clicks the "Game Library" button or menu option.
2. The system displays a list of available games with brief descriptions and game icons.
3. The user browses through the list and reads the descriptions.
4. The user selects a game to view more detailed information or start playing.

**Post Conditions**: The user views the list of available games and can choose a game to play or explore further.

**Exceptions**:

* The game library fails to load due to server or network issues.
* No games are available in the library (e.g., system maintenance or error).

**Priority**: High priority, as it provides access to the main functionality of playing games.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Multiple times per session, as needed by the user.

**Channel to Actor**: User interacts with the "Game Library" button or menu option.

**Secondary Actors**: N/A

**Open Issues**: N/A

**Use Case: Search by ID**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to search for other players using a unique player ID to view their profiles or initiate interactions, such as challenges.

**Preconditions**: The user must be logged in and on the main menu, profile screen, or a page that includes the search functionality.

**Trigger**: The user enters a unique player ID into the search bar and initiates the search.

**Scenario**:

1. The user navigates to the search bar and enters a player ID.
2. The user clicks the "Search" button or presses "Enter."
3. The system verifies the ID and retrieves the associated player profile.
4. The system displays the player’s profile, showing relevant statistics, rank, and other information.
5. The user can view the profile or initiate interactions, such as sending a challenge or viewing game history.

**Post Conditions**: The user successfully views the profile of the player associated with the entered ID and can interact with them.

**Exceptions**:

* No player profile is found for the entered ID.
* The search fails due to network issues or server errors.

**Priority**: Medium priority, as it provides an efficient way to locate specific players.

**When Available**: Within its 1st iteration.

**Frequency of Use**: As needed by the user to locate specific players or profiles.

**Channel to Actor**: User interacts with the search bar and clicks the "Search" button or presses "Enter."

**Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Search by Name**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Allows the user to search for other players by name to view their profiles or challenge them to a game.

**Preconditions**: The user must be logged in and on the main menu, profile screen, or a page that allows access to the search functionality.

**Trigger**: The user enters a name into the search bar and initiates the search.

**Scenario**:

1. The user navigates to the search bar and enters the name of a player.
2. The user clicks the "Search" button or presses "Enter."
3. The system retrieves and displays a list of matching players based on the entered name.
4. The user selects a player from the search results.
5. The system displays the selected player's profile or provides options to interact, such as challenging them to a game.

**Post Conditions**: The user successfully views the search results and can access a player's profile or initiate an interaction.

**Exceptions**:

* No matching player profiles found.
* The search fails due to network issues or server errors.

**Priority**: Medium priority, as it enhances user interaction and competitive gameplay.

**When Available**: Within its 1st iteration.

**Frequency of Use**: As needed by the user to find opponents or view profiles.

**Channel to Actor**: User interacts with the search bar and clicks the "Search" button or presses "Enter."

**Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Show Win Alert for Connect4**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Notifies the user when a player wins the Connect4 game.

**Preconditions**: The user must be playing a game, and a win condition must be detected.

**Trigger**: The system detects four consecutive discs of the same color vertically, horizontally, or diagonally.

**Scenario**:

1. The system checks for a winning condition after each move.
2. If a win is detected, an alert is shown with the winning player's name.
3. The user is given options to either play again or exit the game.
4. If the user chooses to play again, the board is reset for a new game.
5. If the user chooses to exit, the application closes or navigates back to the main menu.

**Post Conditions**: The user is informed of the game result and can choose to continue or exit.

**Exceptions**:

* The alert fails to display due to an internal error.

**Priority**: High priority, as it is crucial for game result notification.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Once per game when a win condition is met.

**Channel to Actor**: System displays an alert on the screen.

**Secondary Actors**: N/A

**Open Issues**: N/A

### **Use Case: Show Win Alert for Checkers**

**Iteration**: 1

**Primary Actor**: User

**Goal in Context**: Notifies the user when a player wins the Checkers game.

**Preconditions**: The user must be playing a game, and a win condition (all opponent’s pieces captured or blocked) must be detected.

**Trigger**: The system detects that one player has no valid moves left or all their pieces have been captured.

**Scenario**:

1. The system checks for a winning condition after each move.
2. If a win is detected, an alert is shown with the winning player's name.
3. The user is given options to either play again or exit the game.
4. If the user chooses to play again, the board is reset for a new game.
5. If the user chooses to exit, the application closes or navigates back to the main menu.

**Post Conditions**: The user is informed of the game result and can choose to continue or exit.

**Exceptions**:

* The alert fails to display due to an internal error.

**Priority**: High priority, as it is crucial for game result notification.

**When Available**: Within its 1st iteration.

**Frequency of Use**: Once per game when a win condition is met.

**Channel to Actor**: System displays an alert on the screen.

**Secondary Actors**: N/A

**Open Issues**: N/A