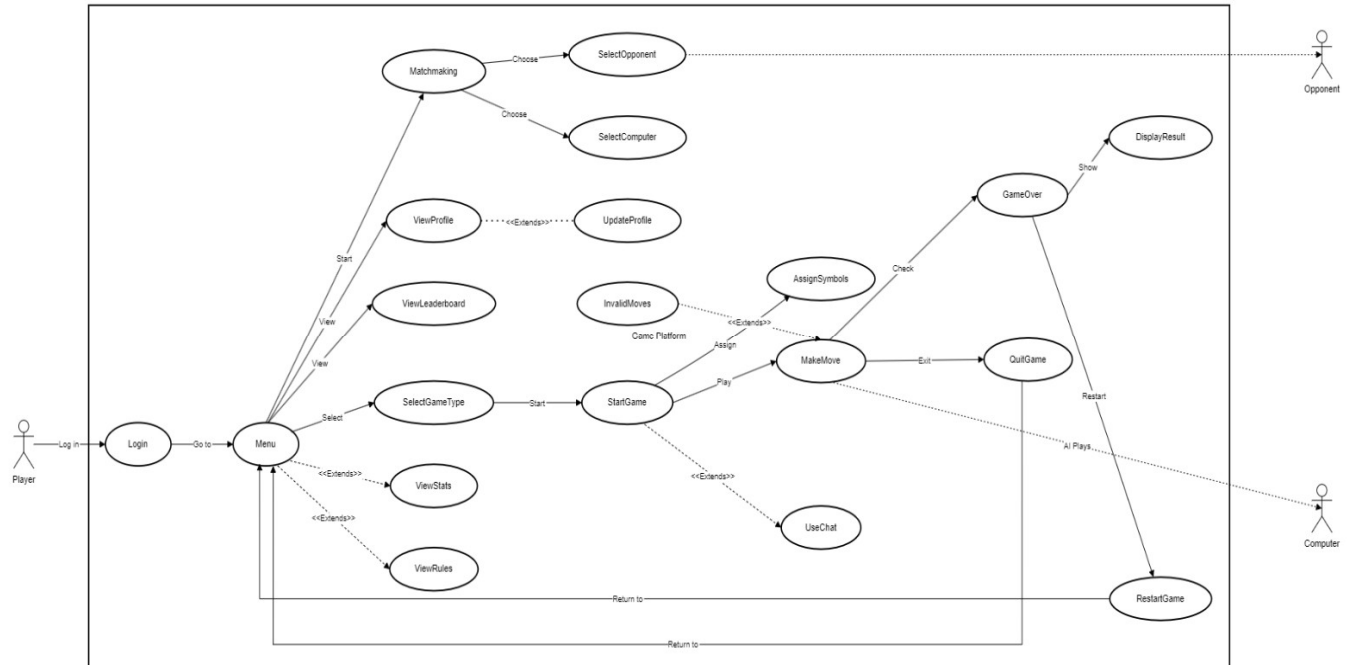


Use Case Diagram (GUI)



Use Case Descriptions (GUI)

Use Case 1: Login

- Iteration: 1, login to the application
- Primary actor: Player
- Goal in context: To authenticate and access the application menu
- Preconditions:
 - The application is running
 - Player has valid login credentials
- Trigger:
 - Player enters login credentials and clicks login
- Scenario:
 - Player launches the application
 - System displays login screen
 - Player enters username and password
 - Player clicks login button

- System validates credentials
 - System grants access to the main menu
- Exceptions:
 - Invalid credentials
 - System unavailable
 - Connection failure
- Priority: High
- When available: First release
- Frequency of Use: Once per session
- Channel to actor: Graphical User Interface
- Secondary actors: Authentication system
- Channels to secondary actors: System API
- Open Issues: Password recovery mechanism

Use Case 2: Access Menu

- Iteration: 1, navigate application menu
- Primary actor: Player
- Goal in context: To navigate between different application features
- Preconditions:
 - Player is logged into the system
- Trigger:
 - Successful login or return from other screens
- Scenario:
 - System displays main menu with available options
 - Player views available menu choices (game type selection, profile, leaderboard, stats, rules)
 - Player selects desired option
 - System navigates to the selected feature
- Exceptions:
 - System error
 - Connection failure
- Priority: High
- When available: First release
- Frequency of Use: Multiple times per session
- Channel to actor: Graphical User Interface
- Secondary actors: None
- Channels to secondary actors: None
- Open Issues: Menu personalization options

Use Case 3: View Profile

- Iteration: 1, access player profile

- Primary actor: Player
- Goal in context: To view personal account information
- Preconditions:
 - Player is at the main menu
- Trigger:
 - Player selects “View Profile” option
- Scenario:
 - Player clicks on View Profile option from menu
 - System retrieves player profile information
 - System displays profile page
 - Player reviews information
 - Player has option to update profile or return to menu
- Exceptions:
 - Data retrieval failure
 - System error
- Priority: Medium
- When available: First release
- Frequency of Use: Occasional
- Channel to actor: Graphical User Interface
- Secondary actors: Database system
- Channels to secondary actors: Data API
- Open Issues: Profile customization options

Use Case 4: Update Profile

- Iteration: 1, modify player profile
- Primary actor: Player
- Goal in context: To change account information or preferences
- Preconditions:
 - Player is viewing their profile
- Trigger:
 - Player initiates update process
- Scenario:
 - Player selects update option
 - System displays editable profile fields
 - Player makes desired changes
 - Player submits changes
 - System validates input
 - System updates profile information
 - System confirms successful update
- Exceptions:
 - Invalid input

- Database update failure
 - System error
- Priority: Medium
- When available: First release
- Frequency of Use: Occasional
- Channel to actor: Graphical User Interface
- Secondary actors: Database system
- Channels to secondary actors: Data API
- Open Issues: Security for sensitive information changes

Use Case 5: View Leaderboard

- Iteration: 1, access player rankings
- Primary actor: Player
- Goal in context: To view rankings of players
- Preconditions:
 - Player is at the main menu
- Trigger:
 - Player selects “View Leaderboard” option
- Scenario:
 - Player clicks on View Leaderboard option from menu
 - System retrieves current rankings
 - System displays leaderboard with player rankings
 - Player reviews information
 - Player returns to menu when finished
- Exceptions:
 - Data retrieval failure
 - System error
- Priority: Low
- When available: First release
- Frequency of Use: Occasional
- Channel to actor: Graphical User Interface
- Secondary actors: Database system
- Channels to secondary actors: Data API
- Open Issues: Filtering options for leaderboard view

Use Case 6: View Stats

- Iteration: 1, access player statistics
- Primary actor: Player
- Goal in context: To view performance statistics
- Preconditions:
 - Player is at the main menu

- Trigger:
 - Player selects “View Stats” option
- Scenario:
 - Player clicks on View Stats option from menu
 - System retrieves player’s statistics
 - System displays detailed statistics
 - Player reviews statistics
 - Player returns to menu when finished
- Exceptions:
 - Data retrieval failure
 - System error
- Priority: Low
- When available: First release
- Frequency of Use: Occasional
- Channel to actor: Graphical User Interface
- Secondary actors: Database system
- Channels to secondary actors: Data API
- Open Issues: Adding more statistical categories

Use Case 7: View Rules

- Iteration: 1, access application rules
- Primary actor: Player
- Goal in context: To view rules and instructions
- Preconditions:
 - Player is at the main menu
- Trigger:
 - Player selects “View Rules” option
- Scenario:
 - Player clicks on View Rules option from menu
 - System displays rules and instructions
 - Player reads rules
 - Player returns to menu when finished
- Exceptions:
 - System error
- Priority: Low
- When available: First release
- Frequency of Use: Occasional
- Channel to actor: Graphical User Interface
- Secondary actors: None
- Channels to secondary actors: None
- Open Issues: Adding tutorial mode

Use Case 8: Select Game Type

- Iteration: 1, choose game configuration
- Primary actor: Player
- Goal in context: To select the desired game type
- Preconditions:
 - Player is at the main menu
- Trigger:
 - Player selects “Select Game Type” option
- Scenario:
 - Player clicks on Select Game Type option from menu
 - System displays available game types
 - Player selects desired game type
 - System proceeds to further game setup steps
- Exceptions:
 - System error
 - Option unavailable
- Priority: High
- When available: First release
- Frequency of Use: Once per game
- Channel to actor: Graphical User Interface
- Secondary actors: None
- Channels to secondary actors: None
- Open Issues: Adding more game types in future updates

Use Case 9: Matchmaking

- Iteration: 1, find an opponent
- Primary actor: Player
- Goal in context: To find and connect with another player or computer
- Preconditions:
 - Player has selected game type
- Trigger:
 - System initiates matchmaking after game type selection
- Scenario:
 - System displays matchmaking screen
 - System processes matchmaking request
 - System presents options to select opponent or computer
 - Player chooses the desired option
- Exceptions:
 - No available matches
 - Connection failure
 - Player cancels matchmaking

- Priority: High
- When available: First release
- Frequency of Use: Once per game
- Channel to actor: Graphical User Interface
- Secondary actors: Matchmaking system
- Channels to secondary actors: Network connection
- Open Issues: Matchmaking algorithms and parameters

Use Case 10: Select Opponent

- Iteration: 1, choose human opponent
- Primary actor: Player
- Goal in context: To select a specific human opponent
- Preconditions:
 - Player has gone through matchmaking
 - Other players are available
- Trigger:
 - Player selects “Select Opponent” option
- Scenario:
 - System displays list of available opponents
 - Player selects desired opponent
 - System initiates connection with selected opponent
 - System proceeds to game start upon successful connection
- Exceptions:
 - Opponent unavailable
 - Connection failure
 - Matchmaking error
- Priority: Medium
- When available: First release
- Frequency of Use: Once per multiplayer game
- Channel to actor: Graphical User Interface
- Secondary actors: Opponent
- Channels to secondary actors: Game system
- Open Issues: Opponent selection criteria

Use Case 11: Select Computer

- Iteration: 1, choose computer opponent
- Primary actor: Player
- Goal in context: To play against computer
- Preconditions:
 - Player has gone through matchmaking
- Trigger:

- Player selects “Select Computer” option
- Scenario:
 - System displays computer options
 - Player selects desired computer settings
 - System configures computer opponent
 - System proceeds to game start
- Exceptions:
 - System error in computer configuration
- Priority: Medium
- When available: First release
- Frequency of Use: Once per computer game
- Channel to actor: Graphical User Interface
- Secondary actors: Computer AI
- Channels to secondary actors: System API
- Open Issues: Computer difficulty levels

Use Case 12: Start Game

- Iteration: 1, initialize game session
- Primary actor: Player
- Goal in context: To begin playing a game
- Preconditions:
 - Player has selected opponent (human or computer)
- Trigger:
 - System automatically initiates after opponent selection
- Scenario:
 - System initializes the game
 - System displays game interface
 - System assigns player roles
 - System enables game play
- Exceptions:
 - Connection failure with opponent
 - System error during initialization
- Priority: High
- When available: First release
- Frequency of Use: Once per game
- Channel to actor: Graphical User Interface
- Secondary actors: Opponent or Computer
- Channels to secondary actors: Game system
- Open Issues: Handling unexpected disconnections

Use Case 13: Assign Symbols

- Iteration: 1, assign players their game symbols
- Primary actor: Game System
- Goal in context: To determine which player uses which symbol
- Preconditions:
 - Game has been started
- Trigger:
 - Game initialization
- Scenario:
 - System assigns symbols to players
 - System displays symbol assignment to players
 - System determines first player
 - System enables first player to make a move
- Exceptions:
 - System error
- Priority: Medium
- When available: First release
- Frequency of Use: Once per game
- Channel to actor: Graphical User Interface
- Secondary actors: Players
- Channels to secondary actors: Game system
- Open Issues: Player symbol preferences

Use Case 14: Make Move

- Iteration: 1, player makes a game move
- Primary actor: Player
- Goal in context: To perform an action in the game
- Preconditions:
 - Game has started
 - It is the player's turn
- Trigger:
 - Player initiates a move
- Scenario:
 - System indicates it's player's turn
 - Player selects their move
 - System validates the move
 - System processes the move
 - System checks for game state changes
 - System passes turn to opponent if game continues
- Exceptions:
 - Invalid move

- Connection issues
- Priority: High
- When available: First release
- Frequency of Use: Multiple times per game
- Channel to actor: Graphical User Interface
- Secondary actors: Game system
- Channels to secondary actors: Game logic
- Open Issues: Move validation rules

Use Case 15: Invalid Moves

- Iteration: 1, handle invalid move attempts
- Primary actor: Game System
- Goal in context: To prevent and notify about invalid moves
- Preconditions:
 - Player attempts an invalid move
- Trigger:
 - System detects an invalid move
- Scenario:
 - Player attempts a move
 - System validates the move
 - System determines move is invalid
 - System displays error message
 - System prompts player to try again
- Exceptions:
 - System error during validation
- Priority: Medium
- When available: First release
- Frequency of Use: As needed
- Channel to actor: Graphical User Interface
- Secondary actors: None
- Channels to secondary actors: None
- Open Issues: Clear communication of move restrictions

Use Case 16: Use Chat

- Iteration: 1, communicate with opponent
- Primary actor: Player
- Goal in context: To send messages during game
- Preconditions:
 - Game has started
 - Playing against human opponent
- Trigger:

- Player accesses chat feature
- Scenario:
 - Player activates chat interface
 - Player types message
 - Player sends message
 - System delivers message to opponent
 - System displays opponent's responses
- Exceptions:
 - Connection failure
 - Chat filtering issues
- Priority: Low
- When available: After multiplayer implementation
- Frequency of Use: Optional during gameplay
- Channel to actor: Graphical User Interface
- Secondary actors: Opponent
- Channels to secondary actors: Chat system
- Open Issues: Chat moderation and filtering

Use Case 17: Game Over

- Iteration: 1, end game and determine result
- Primary actor: Game System
- Goal in context: To conclude the game
- Preconditions:
 - Game end condition has been reached
- Trigger:
 - System detects game end condition
- Scenario:
 - System detects game end
 - System stops accepting further moves
 - System transitions to game over state
 - System prepares to show result
 - System enables post-game options
- Exceptions:
 - System error during end-game processing
- Priority: High
- When available: First release
- Frequency of Use: Once per completed game
- Channel to actor: Graphical User Interface
- Secondary actors: Players
- Channels to secondary actors: Game notification system
- Open Issues: Game state preservation

Use Case 18: Display Result

- Iteration: 1, show game outcome
- Primary actor: Game System
- Goal in context: To show the final result to players
- Preconditions:
 - Game is over
 - Result has been determined
- Trigger:
 - System automatically after game over
- Scenario:
 - System retrieves final game state
 - System calculates final result
 - System displays result to players
 - System updates statistics
 - System shows post-game options
- Exceptions:
 - Display error
 - Statistics update failure
- Priority: High
- When available: First release
- Frequency of Use: Once per completed game
- Channel to actor: Graphical User Interface
- Secondary actors: Database system
- Channels to secondary actors: Data API
- Open Issues: Result visualization improvements

Use Case 19: Quit Game

- Iteration: 1, exit current game
- Primary actor: Player
- Goal in context: To end the current game session
- Preconditions:
 - Player is in an active game
- Trigger:
 - Player selects “Quit Game” option
- Scenario:
 - Player selects quit option
 - System confirms player’s intention
 - System processes game termination
 - System updates necessary records
 - System returns player to main menu
- Exceptions:

- Connection failure
 - System error
- Priority: Medium
- When available: First release
- Frequency of Use: As needed
- Channel to actor: Graphical User Interface
- Secondary actors: Opponent (if multiplayer)
- Channels to secondary actors: Game system notification
- Open Issues: Handling early termination consequences

Use Case 20: Restart Game

- Iteration: 1, start a new game with same settings
- Primary actor: Player
- Goal in context: To quickly start another game with the same settings
- Preconditions:
 - Game has ended
- Trigger:
 - Player selects “Restart Game” option
- Scenario:
 - Player selects restart option
 - System confirms with opponent (if multiplayer)
 - System resets the game
 - System maintains same player assignments
 - System begins new game
- Exceptions:
 - Opponent declines restart
 - Connection failure
- Priority: Medium
- When available: First release
- Frequency of Use: Occasional
- Channel to actor: Graphical User Interface
- Secondary actors: Opponent (if multiplayer)
- Channels to secondary actors: Game system
- Open Issues: Maximum number of consecutive restarts