Games: connec4, checkers, whist

Template:

Use Case Title
Use Case:
Iteration:
Primary Actor:
Goal in Context:
Preconditions:
Trigger:

Scenario:
    1.

Postcondition:
Exceptions:
    1.

Priority:
When available:
Frequency of Use:
Channel to actor:
Secondary Actors:
Channel to secondary Actors:
Open Issues:

Actors
Primary users:
Active Player: the player whose turn it is
Passive Player: the other player
Software actors:
Network
Database
User Interface
Game host server

Initiating a game

Use Case:
Iteration: 1
Primary Actor: Player
Goal in Context: Starting a game with another player
Preconditions: The player is logged in
Trigger: The player selects a game and clicks play

Scenario:
1. The game matches another player of similar skill level
2. An object of the game selected is created
3. The UI displays the game board to both players

Postcondition: the selected game is ready to play for both players
Exceptions:
1. The player is matched to themself

Priority: high
When available: initial release
Frequency of Use: high, once every game
Channel to actor: network UI update
Secondary Actors: game host server
Channel to secondary Actors: network
Open Issues:

<u>Dropping a Connect Four Checker</u>
Use Case:
Iteration: 1
Primary Actor: Active player
Goal in Context: The active player dropping their checker, completing their turn
Preconditions: A valid game is active
Triggers: The beginning of the game and the end of a turn that doesn't result in the end of the game.

Scenario:
1. A game is initiated and validated
2. Player 1 makes the first move

<u>Postcondition: the move reflected in the UI and displayed to both players</u>
Exceptions:
1. The game is not valid, for example, a player is matched with themself.
2. A player quits or disconnects, ending the game.

Priority: high, this is the core of the Connect Four gameplay.
When available: first release.
Frequency of Use: high, used multiple times every game.
Channel to actor: network UI update
Secondary Actors: Passive player, game host server
Channel to secondary Actors: network
Open Issues:

Ending a game

Use Case:
Iteration: 1
Primary Actor: active player
Goal in Context: displaying to all players that the game is over, and if they lost, won or drew.
Preconditions: A game is active
Trigger: A wining, losing, or drawing move, or a player quits

Scenario:
1. The server detects that the game is over
2. The players' stats are updated
3. The UI displays to all players whether they won, drew or lost
4. The UI displays the option to leave the game

Postcondition: The game is over
Exceptions: The game will always end

Priority: high
When available: first release
Frequency of Use: High, used every game
Channel to actor: network
Secondary Actors: passive players, game logic server
Channel to secondary Actors: network
Open Issues: