Use Case Descriptions

Use Case: Choose Game

Iteration: 1

Primary Actor: Player

Goal in Context: The player wants to select which games to play (Tic Tac Toe, Connect Four, or Checkers)

Preconditions
- The player is logged in.
- The main menu is displayed with available game options.

Trigger: The player navigates to the main game selection screen.

Scenario:
1. The system shows a list of available games (Tic Tac Toe, Connect Four, Checkers).
2. The player selects one of the games.
3. The system transitions to that game's interface.

Postconditions: The chosen game is now active and the player an proceed to join a queue.

Exceptions:
- Game list fails to load due to system error.
- Chosen game is temporarily unavailable.

Priority: Medium

When Available: Always

Frequency of Use: High (everytime a player wants to switch or start a new game)

Channel to Actor: GUI

Secondary Actors: Database

Channel to Secondary Actors: N/A

Open Issues
- How do we handle showing the GUI when a game is under maintenance?

Use Case: Enter Queue

Iteration: 1

Primary Actor: Player

Goal in Context: The player wants to find an opponent by joining the matchmaking queue for the chosen game.

Preconditions
- The player has selected a game (Tic Tac Toe, Connect Four, or Checkers)
- The player has a valid MMR rank.

Trigger: The player clicks the “Enter Queue” or “Join Queue” button.

Scenario:
1. The system checks the player’s rank and MMR for the chosen game.
2. The system places the player in the corresponding matchmaking queue (or queue pair).
3. The player sees a confirmation that they have joined the queue.

Postconditions: The player is now in the matchmaking queue, awaiting an opponent.

Exceptions:
- The player is already in a queue.

Priority: High

When Available: Always

Frequency of Use: High (everytime a player wants to start a match)

Channel to Actor: GUI

Secondary Actors: Matchmaking Service

Channel to Secondary Actors: N/A

Open Issues

- How do we handle a large number of concurrent queue requests?