CHECKERS

Use Case: Player joins matchmaking queue

Iteration: 1

Primary Actor: Player

Goal in Context: The player wants to quickly find an opponent by entering the Checkers matchmaking system.

Preconditions:

- The player is logged in.
- The player has a valid account and an assigned MMR/rank for checkers.

Trigger: The player selects the “Join Queue” option from the Checkers game interface.

Scenario:

1. The player clicks the “Join Queue” button.
2. The system verifies the player’s current Checkers rank and MMR.
3. The system randomly assigns the player to one of the two queue pairs for their rank.
4. The player is added to the selected Checkers matchmaking queue.

Postconditions: The player is now waiting in a Checkers matchmaking queue for pairing.

Exceptions:

- Player is already in an active queue.
- Network/server issues prevent the queue join.

Priority: High

When Available: Always

Frequency of Use: High (each game session)

Channel to Actor: Checkers client interface

Secondary Actors: Checkers matchmaking service

Channel to Secondary Actors: N/A

Open Issues:

- Determining how to handle simultaneous queue join requests.

Use Case: MMR changes after a game completes

Iteration: 1

Primary Actor: Game/Player

Goal in Context: Update the player's MMR based on the outcome of a completed Checkers game.

Preconditions:

- The Checkers game has just finished.
- The game outcome (win, loss, or tie) is recorded.

Trigger: The Checkers game has ended.

Scenario:

1. The game computes the result of the Checkers game.
2. The system invokes the corresponding method on the player's Checkers stats object.
3. The player's MMR is recalculated.
4. Updated MMR is stored and reflected in the player's profile and leaderboard.

Postconditions: The player's Checkers MMR accurately reflects their recent game performance.

Exceptions:

- Errors in MMR calculation.

Priority: High

When Available: Always

Frequency of Use: Every Checkers game completion.

Channel to Actor: Internal game processing.

Secondary Actors: Stats and leaderboard

Channel to Secondary Actors: N/A

Open Issues:

- Refine the formula used to update player ratings in Checkers so that changes are smooth and fairly reflect performance differences.

Use Case: Display Leaderboard

Iteration: 1

Primary Actor: Game/Player

Goal in Context: The player wants to view a ranked list of Checkers players based on metrics such as MMR or wins.

Preconditions:

- The Checkers leaderboard data is available.
- The game is responsive.

Trigger: The player selects the “View Leaderboard” option from the Checkers game menu.

Scenario:

1. The player clicks on the “Leaderboard” tab.
2. The system retrieves Checkers leaderboard data.
3. The sorted leaderboard is displayed on the player's screen.

Postconditions: The player is presented with an updated and ranked leaderboard for Checkers.

Exceptions:

- Missing leaderboard stats.

Priority: High

When Available: Always

Frequency of Use: Moderate

Channel to Actor: Checkers game client interface

Secondary Actors: Leaderboard system

Channel to Secondary Actors: N/A

Open Issues:

- How will we ensure the stats are calculated effeciently and that the display of the leaderboard is consistent even with large data sets?