Matchmaking System - Use Case Descriptions

Use Case 1: Find a Match

Actors:

- Player (Primary)
- Matchmaking System (Supporting)

Preconditions:

- The player is logged in.
- The matchmaking system is operational.

Trigger:

- The player selects “Find Match” from the UI.

Main Flow:

1. The Player requests a match.
2. The Matchmaking System retrieves the player’s skill level and search criteria.
3. The system searches for an available opponent within a similar skill range.
4. If a match is found:
    - The system pairs both players.
    - A match confirmation is sent to both players.
    - The game session begins.
5. If no match is found:
    - The player is placed in a queue.
    - The system reattempts matchmaking periodically.

Alternative Flows:

- **Opponent Not Found:** The system expands the skill range after a timeout.
- **Player Cancels Matchmaking:** The system removes the player from the queue.
- **Player Disconnects While Searching:** The player is removed from matchmaking.

Postconditions:

- A new match is created and started.
- If no match is found, the player remains in the queue.

Use Case 2: Player Joins a Match

Actors:

- Player (Primary)
- Matchmaking System (Supporting)
- Opponent Player (Supporting)

Preconditions:

- The player is logged in.
- The player has been matched with an opponent.

Trigger:

- The matchmaking system finds an opponent and sends a match confirmation.

Main Flow:

1. The Matchmaking System notifies both players of a found match.
2. Each Player accepts the match.
3. The Matchmaking System starts the game session.

Alternative Flows:

- **Player Rejects Match:** The match is canceled, and the system searches for a new opponent.
- **Opponent Declines:** The system searches for another match.
- **Connection Issues:** If a player disconnects before confirming, the system cancels the match.

Postconditions:

- If both players accept, the match starts.
- If a player rejects, matchmaking continues.

Use Case 3: Queue System for Matchmaking

Actors:

- Player (Primary)
- Matchmaking System (Supporting)

Preconditions:

- The player is logged in.
- The player has requested a match.

Trigger:

- No immediate opponent is found.

Main Flow:

1. The Matchmaking System places the player in a queue.
2. The system periodically searches for new opponents.
3. If an opponent is found:
    - The system pairs them.
    - A match confirmation is sent.
4. If no match is found after a set time:
    - The system expands the search criteria.
    - The system may provide an option to exit the queue.

Alternative Flows:

- **Player Leaves Queue:** The player manually exits matchmaking.
- **Opponent Found Late:** If a match takes too long, the player may be given an option to cancel.

Postconditions:

- The player either gets matched or remains in the queue.

Use Case 7: Start a Match

Actors:

- Player 1 (Primary)
- Player 2 (Primary)
- Matchmaking System (Supporting)
- Game Logic System (Supporting)

Preconditions:

- The matchmaking system has found a valid opponent.
- Both players have accepted the match.

Trigger:

- The matchmaking system successfully pairs two players.

Main Flow:

1. The Matchmaking System selects the game type based on player preferences.
2. The system creates a new game session and assigns Player 1 and Player 2.
3. The Game Logic System loads the game rules and initializes the board.
4. The system notifies both players that the match is ready.
5. The game officially begins, and turns are assigned.

Alternative Flows:

- **One player disconnects before starting:** The match is canceled, and the remaining player returns to matchmaking.
- **Match fails to initialize:** Both players return to matchmaking.

Postconditions:

- The game has officially started.
- Both players are engaged in the game session.

Use Case 8: Update Matchmaking After a Match (Wins, Losses, and Statistics)

Actors:

- Player 1 (Primary)
- Player 2 (Primary)
- Matchmaking System (Supporting)
- Game Logic System (Supporting)

Preconditions:

- The game has ended.
- The Game Logic System has determined the winner and loser (or a draw).

Trigger:

- The game reaches a win condition, loss condition, or draw.

Main Flow:

1. The Game Logic System identifies the match outcome:
    - Winner is determined.
    - Loser is recorded.
    - If applicable, a draw is declared.
2. The Matchmaking System updates player statistics:
    - Increase the winner's win count.
    - Increase the loser's loss count.
    - Update the number of games played for both players.
3. The system updates matchmaking ranking (if skill-based matchmaking applies).
4. Both players receive a summary of their updated stats.

Alternative Flows:

- **A player disconnects before the match ends:** The system awards a win by forfeit to the remaining player.
- **Match results fail to update due to an error:** The system retries updating the stats.

Postconditions:

- The winner and loser stats are updated.
- The matchmaking system reflects updated skill levels or ranks.

Use Case 9: Player Requests a Rematch

Actors:

- Player 1 (Primary)
- Player 2 (Primary)
- Matchmaking System (Supporting)

Preconditions:

- A match has ended.
- Both players are still online.

Trigger:

- One player selects “Request Rematch.”

Main Flow:

1. The Player sends a rematch request to the opponent.
2. The Matchmaking System notifies the other player.
3. If the opponent accepts, a new match is created with the same players.
4. If the opponent declines, the requesting player returns to matchmaking.

Alternative Flows:

- **The opponent declines:** The player is placed back into normal matchmaking.
- **One player disconnects before responding:** The request is automatically canceled.

Postconditions:

- A rematch is started if both players agree.
- If declined, players are returned to regular matchmaking.