

Use Case 1: Find a Match

Actors:

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

Preconditions:

- The player is logged into their account.
- 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 with a similar skill level.
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.