**Define “max” as the maximum number of players who can play one game of Medieval Warfare together.**

# 1: Use Cases

## Summary Use Case

**Use Case:** Win Game

**Scope:** Game Client System

**Level:** Summary

**Intention in context:** The *User* wants to win a game of Medieval Warfare.

**Multiplicity:** Many *Users* may perform this action, each sequentially.

**Primary Actor:** User

**Main Success Scenario:**

1. *User* sets up game for play.
2. *User* plays game.

**Extensions:**

1a. Setting up game results in failure; use case results in failure.

2a. Playing game results in failure; use case results in failure.

## Game Setup Use Cases

**Use Case:** Set up Game for Play

**Scope:** Game Client System

**Level:** User Goal

**Intention in context:** The *User* wants to set up a playable game of Medieval Warfare.

**Multiplicity:** Many *Users* may perform this action, each sequentially, and requires 1 to max - 1 other users partaking in the later stages of setup of that same game.

**Primary Actor:** User

**Secondary Actor:** Matchmaking Server, other Players, game Host (if not User)

**Main Success Scenario:**

1. *User* logs in.
2. *User* joins a game lobby or creates a game lobby.
3. *User* configures the game.
4. *User* informs *System* that he is ready.
5. All *Players* (of which there are at least 2) have informed *System* of their readiness.

**Extensions:**

(1-7)a. *System* fails to establish contact with *Matchmaking Server*.

(1-7)a.1. *System* informs *User* of disconnection; use case ends in failure.

(2-5)a. *User* informs *System* that he wants to return to the previous step.

(2-5)a.1. *System* returns *User* to previous step; use case resumes at previous step.

(4-7)a. *Host* leaves lobby.

(4-7)a.1. *System* removes *User* from lobby; use case resumes at step 2.

1a. *User* does not have an account.

1a.1. *User* creates an account; the use case continues at step 1.

1b. *User* requests *System* to exit.

1b.1. *System* closes; use case ends in failure.

2a. *User* informs *System* of his desire to view a player’s statistics.

2a.1. *System* gives *User* that player’s statistics; use case continues at step 2.

2b. Joining a game lobby or creating a new game failed; use case continues at step 2.

3a. Configuration use case ends in failure.

3a.1a. The game already has a valid configuration.

3a.1a.1. *System* informs *User* of failure; use case resumes at step 4.

3a.1b. The game does not have a valid configuration.

3a.1b.1. *System* informs *User* of failure; use case continues at step 3.

4a. *User* informs *System* he wishes to change the game configuration again.

4a.1. *System* unsets ready status of all players in lobby; use case resumes at step 3.

6a. User informs *System* he wishes to unready himself.

6a.1a. Not all players in lobby are ready.

6a.1a.1. *System* unsets ready status of *User*; use case continues at step 5.

6a.1b. All players in lobby are ready.

6a.1b.1. *System* informs user of failure to unready; use case continues at step 6.

6b. There are less than two *Users* in the lobby; use case continues at step 6.

**Use Case:** Log In

**Scope:** Game Client System

**Level:** Subfunction

**Intention in context:** *User* wishes to log in to matchmaking server.

**Multiplicity:** Many *Users* may perform this action, but only once per instance of the game running, per account.

**Primary Actor:** User

**Secondary Actor:** Matchmaking Server

**Main Success Scenario:**

1. *User* gives *System* his login credentials.
2. *System* requests *Matchmaking Server* to verify identification information.
3. *Matchmaking Server* informs *System* that credentials are valid.
4. *System* informs *User* of successful login.

**Extensions:**

3a. *Matchmaking Server* informs *System* that credentials are incorrect.

3a.1. *Server* informs *User* and prompts him/her to retry; use case resumes at step 1.

**Use Case:** Create Account

**Scope:** Game Client System

**Level:** Subfunction

**Intention in context:** *User* wishes to create account used for log in.

**Multiplicity:** Many *Users* may perform this action, each sequentially.

**Primary Actor:** User

**Secondary Actor:** Matchmaking Server

**Main Success Scenario:**

1. *User* provides *System* his login credentials (username and password).
2. *System* requests *Matchmaking Server* to verify credentials.
3. *Matchmaking Server* informs *System* that credentials are valid.
4. *System* informs *User* of successful account creation.

**Extensions:**

3a. *Matchmaking Server* informs *System* that username is taken or invalid.

3a.1. *Server* informs *User* and prompts him/her to retry; use case resumes at step 1.

4a. *Matchmaking Server* informs *System* that password is invalid.

4a.1. *Server* informs *User* and prompts him/her to retry; use case resumes at step 1.

**Use Case:** Join Game Lobby

**Scope:** Game Client System

**Level:** Subfunction

**Intention in context:** *User* wishes to join a game which has already been created by another player, who is the *Host* of that lobby / game.

**Multiplicity:** Many *Users* may perform this action, each sequentially, but up to max – 1 players per lobby.

**Primary Actor:** User

**Secondary Actor:** Matchmaking Server, Host

**Main Success Scenario:**

1. *User* requests *System* to join a lobby.
2. *System* requests *Matchmaking Server* to get lobby information.
3. *Matchmaking Server* informs *System* that lobby is not full.
4. *System* places *User* in lobby.

**Extensions:**

3a. *Matchmaking Server* informs *System* that lobby is full.

3a.1. *System* informs *User* about error; use case ends in failure.

3b. *Matchmaking Server* informs *System* that lobby to be entered no longer exists.

3b.1. *System* informs *User* about error; use case ends in failure.

**Use Case:** Host Game Lobby

**Scope:** Game Client System

**Level:** Subfunction

**Intention in context:** *User* wishes to create a new lobby in which he is the host.

**Multiplicity:** Many *Users* may perform this action, each sequentially.

**Primary Actor:** User

**Secondary Actor:** Matchmaking Server

**Main Success Scenario:**

1. *User* requests *System* to create a new lobby.
2. *System* informs *Matchmaking Server* of intent to create a new lobby.
3. *Matchmaking Server* informs *System* that it is cleared to create a new lobby with the *User* as host.
4. *System* places *User* in new lobby as host.

**Extensions:**

3a. *Matchmaking Server* informs *System* that it is not cleared to create a new lobby.

3a.1. *System* informs *User* about error; use case ends in failure.

**Use Case:** Configure Game

**Scope:** Game Client System

**Level:** Subfunction

**Intention in context:** *User* wishes to configure the game settings and map; this includes configuring the game as a continuation of a previously saved game.

**Multiplicity:** Many *Hosts* may perform this action, each sequentially for their own lobbies.

**Primary Actor:** User

**Main Success Scenario:**

1. *User* performs one of the following actions:
   1. Request *System* change game settings.
   2. Request *System* change game map.
   3. Request *System* load a game from a save.
   4. Request *System* clear all settings and any loaded game state.
2. System notifies other connected Users in the lobby of the changes.
3. *System* informs *User* of game state change success.

**Extensions:**

1.1a. Game has been loaded from save.

1.1a.1. *System* informs *User* of error; use case resumes at step 1.

1.2a. Game has been loaded from save.

1.2a.1. *System* informs *User* of error; use case resumes at step 1.

1.3a. *System* is unable to load the game save file specified by *User*.

1.3a.1. *System* informs *User* of error; use case resumes at step 1.

2a A Player connected to the server isn’t successfully notified.

2a.1 The Player is kicked.

2a.2 The User is notified.

## Game Play Use Cases

**Use Case:** Play Game

**Scope:** Game Client System

**Level:** User Goal

**Intention in context:** The *User* wants to play a game of Medieval Warfare to win.

**Multiplicity:** Many *Users* may perform this action, each sequentially, up to max players per game.

**Primary Actor:** User

**Secondary Actor:** Host, Matchmaking server

**Main Success Scenario:**

1. *User* plays a turn.
2. *Host* informs *System* that *User* has won.
3. *System* informs *User* that he/she has won.
4. *System* informs Matchmaking server of all statistics accumulated during the game.

**Extensions:**

(1-3)a. *System* fails to establish contact with *Matchmaking Server*.

(1-3)a.1. *System* informs *User* of disconnection; use case ends in failure.

1a. *User* fails to play a turn; use case ends in failure.

2a. *Host* informs *System* that *User* has not won; use case resumes at step 1.

2b. Host informs System that User has lost.

2b.1. System informs User of lost game; use case ends in failure.

**Use Case:** Play a Turn

**Scope:** Game Client System

**Level:** Subfunction

**Intention in context:** The *User* wants to play a turn of Medieval Warfare.

**Multiplicity:** Many *Users* may perform this action, each sequentially, and for players in the same game, one after the other, up to max players per game.

**Primary Actor:** User

**Secondary Actor:** Host, Players

**Main Success Scenario:**

1. *User* performs any number of the following actions:
   1. Inform *System* of Move.
      1. *System* validates Move.
      2. *System* requests *Host’s* approval of Move.
      3. *Host* approves Move.
      4. *System* makes the move and outputs the updated game board.
   2. Request *System* to save the game.
      1. *System* informs *User* of save success.
2. *User* requests *System* to end the turn.
3. *Host* informs all other *Players* of the game state.
4. *System* informs *User* of game state changes.

**Extensions:**

1.1.1a. *System* fails to validate Move.

1.1.1a.1. *System* informs *User* of failure; use case resumes at step 1.

1.1.3a. *Host* fails to approve Move.

1.1.3a.1. *System* informs *User* of failure; use case resumes at step 1.

1.2a System fails to save game.

1.2a.1. *System* informs *User* of failure; use case resumes at step 1.

2a. *User* requests *System* to leave game; use case ends in failure.