

Network Protocol

Andrea Callari, Simone Checchia, Davide Ciamacco, Federica Currò Dossi

Gruppo AM03

This document presents the network protocol between client and server.
The messages mentioned in the document have the following general syntax:

Observers note

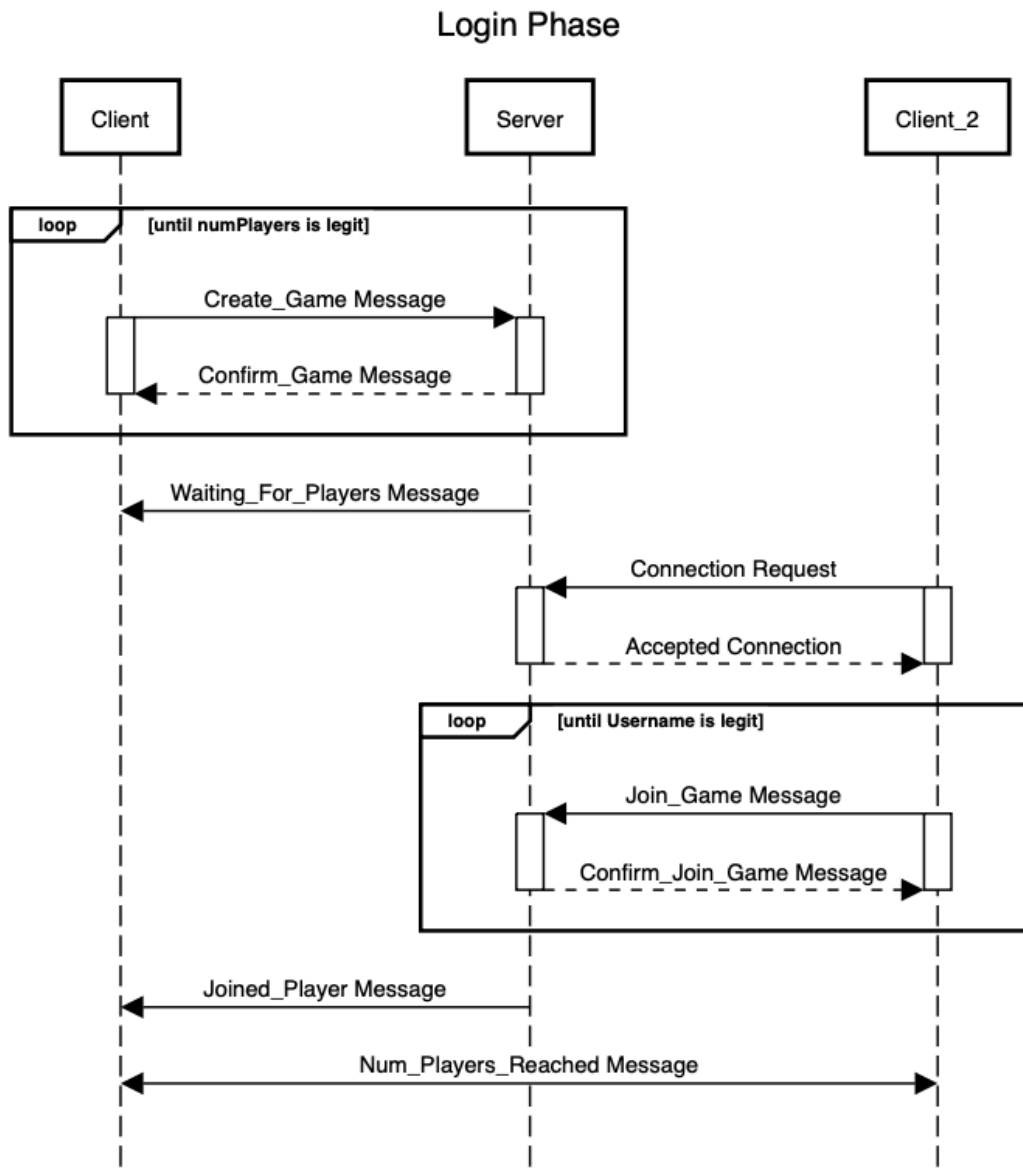
In terms of managing update notification, we will use the Observer design pattern. Each observer will update its subscribers when the class it observes undergoes changes. In the model, there will be four types of observers:

- Player Observer
- Game Observer
- Player_Board Observer
- Chat Observer

All messages in the document starting with 'Notify' or 'Update' will be handled by the Observers.

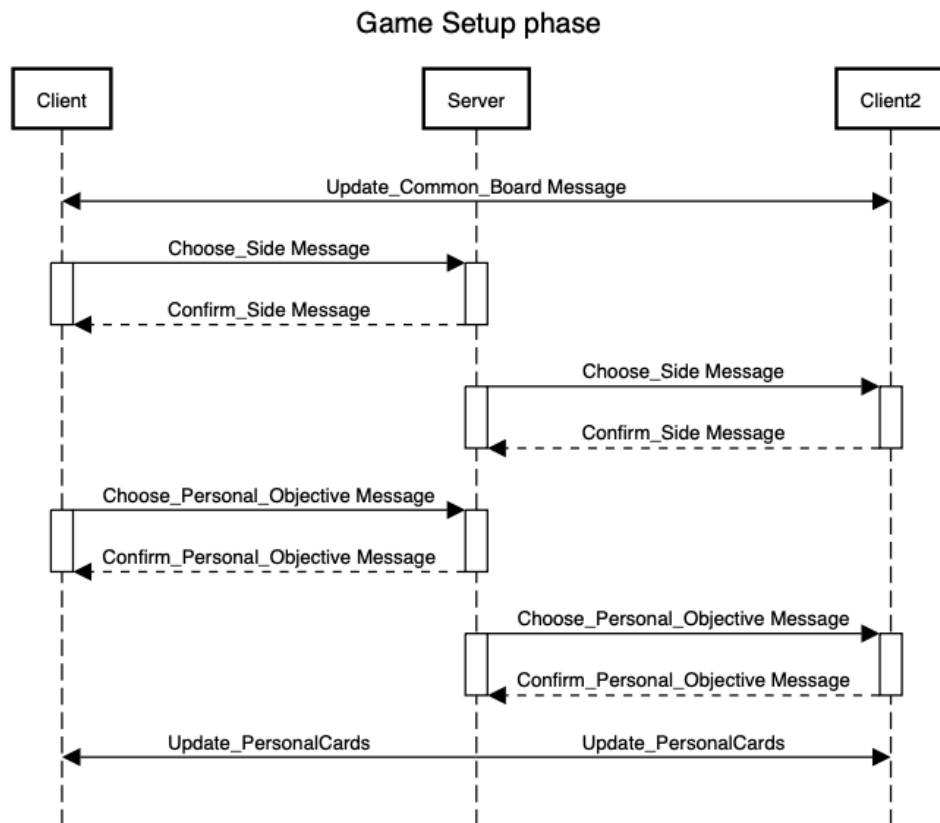
We will illustrate the communication flow through the different phases of the game.

Login



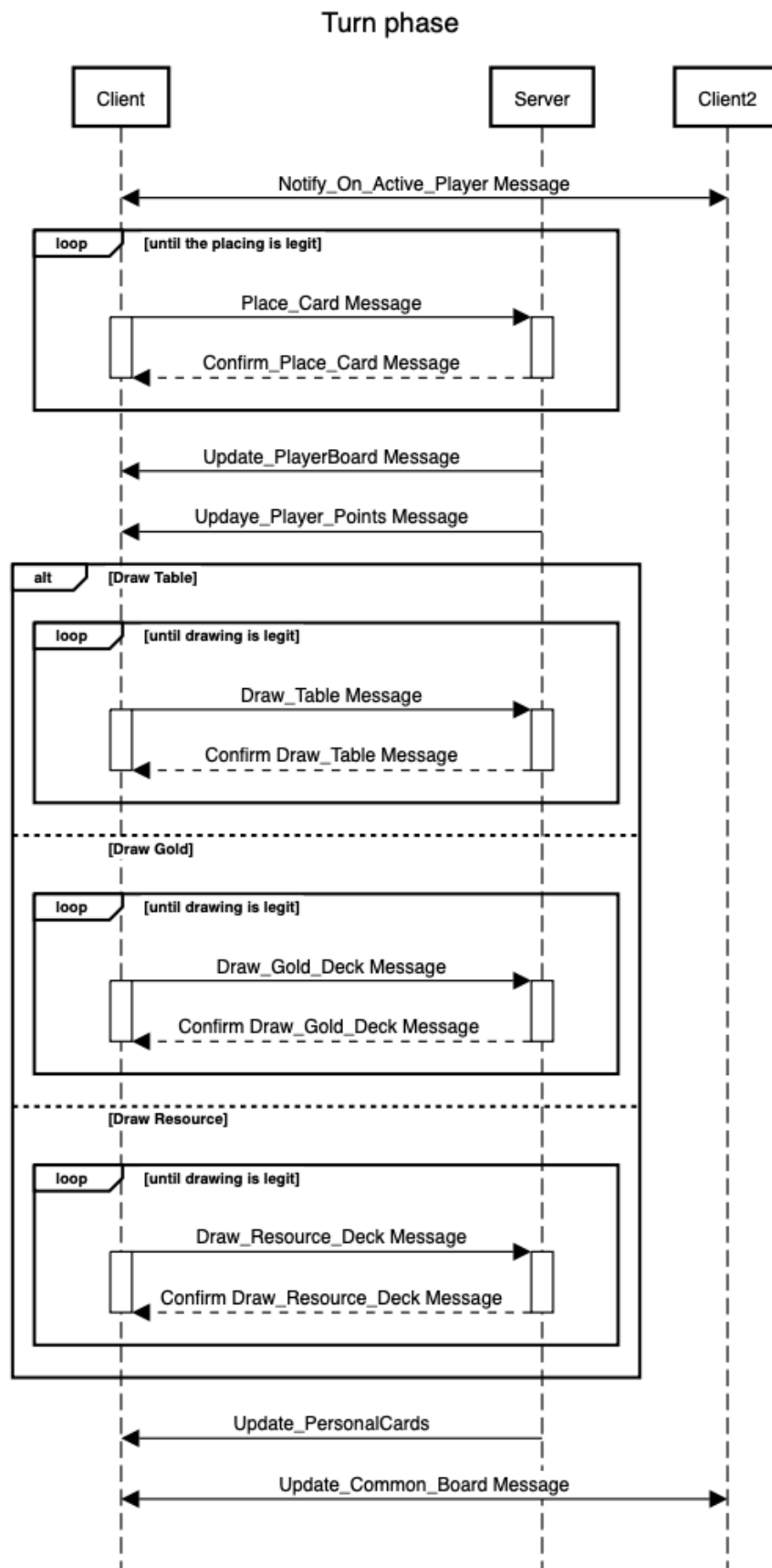
- CreateGameMessage (int numPlayers, String nickname):
The game controller calls .createGame(int numPlayers, String nickname) which creates an instance of Game Model. The request is denied if :
 - 'numPlayers' is <2 or >4
 - nickname is a blank space
- ConfirmGameMessage (Boolean confirmGameCreation, String errorDetails).
- WaitingforPlayersMessage (String message).
- JoinedPlayerMessage (String message).
- NumPlayersReachedMessage (String message).

Game Set-Up



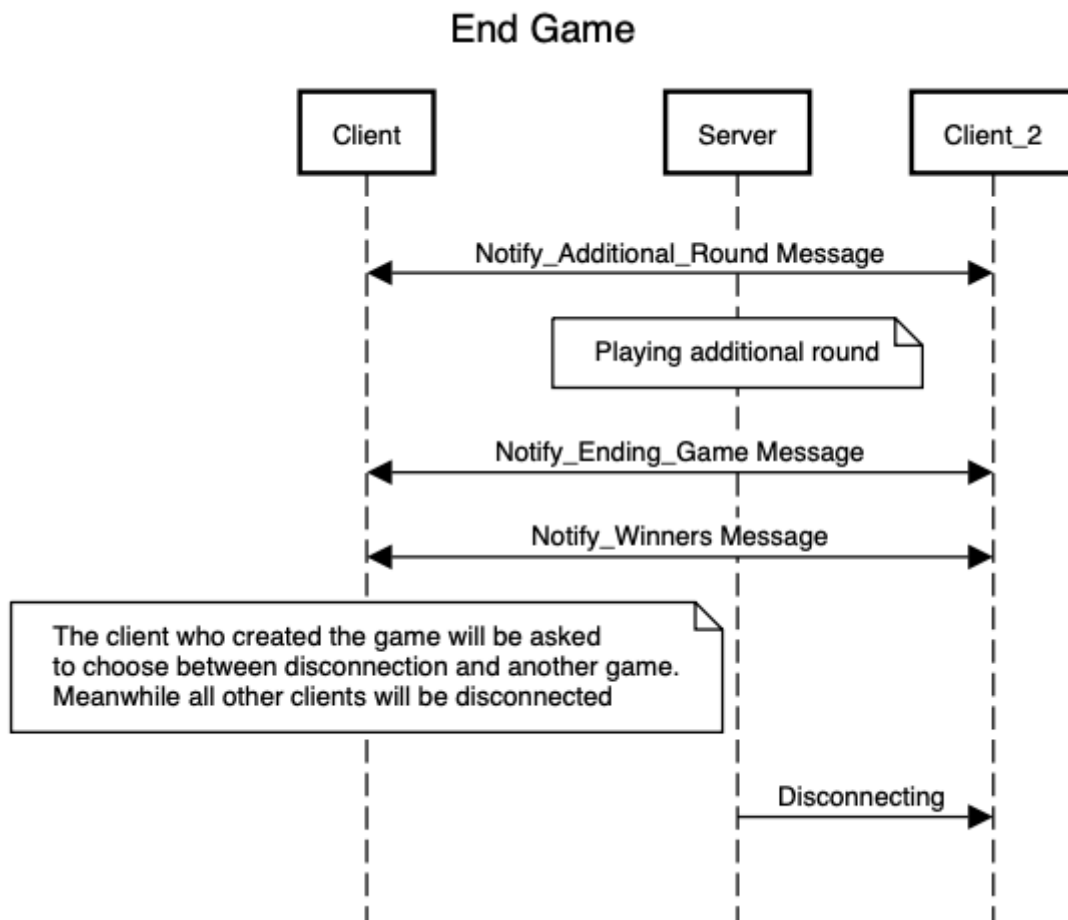
- **UpdateCommonBoardMessage** (ResourceDeck r, GoldDeck ArrayList<PlayableCards> p).
- **ChooseSideMessage** (String nickname, boolean face):
The method called in Game Controller is `selectStartingFace`, which calls on Game Model the method `.selectStartingFace(player, face)`. If the player isn't the active one and the Game State isn't "PREPARATION_1" the request will be denied.
- **ConfirmSideMessage** (Boolean confirmChoice, String errorDetails)
- **ChoosePersonalObjectiveMessage** (String nickname, int choice):
The method called in Game Controller is `setObjectiveCard(String nickname, int choice)` which calls on Game Model the method `.setObjectiveCard(nickname, choice)`. If the player isn't the active one and the Game State isn't "PREPARATION_2" the request will be denied.
- **ConfirmPersonalObjectiveMessage** (Boolean confirmChoice, String errorDetails).
- **UpdatePersonalCardsMessage** (String player, ArrayList <PlayableCards> a).

Turn phase



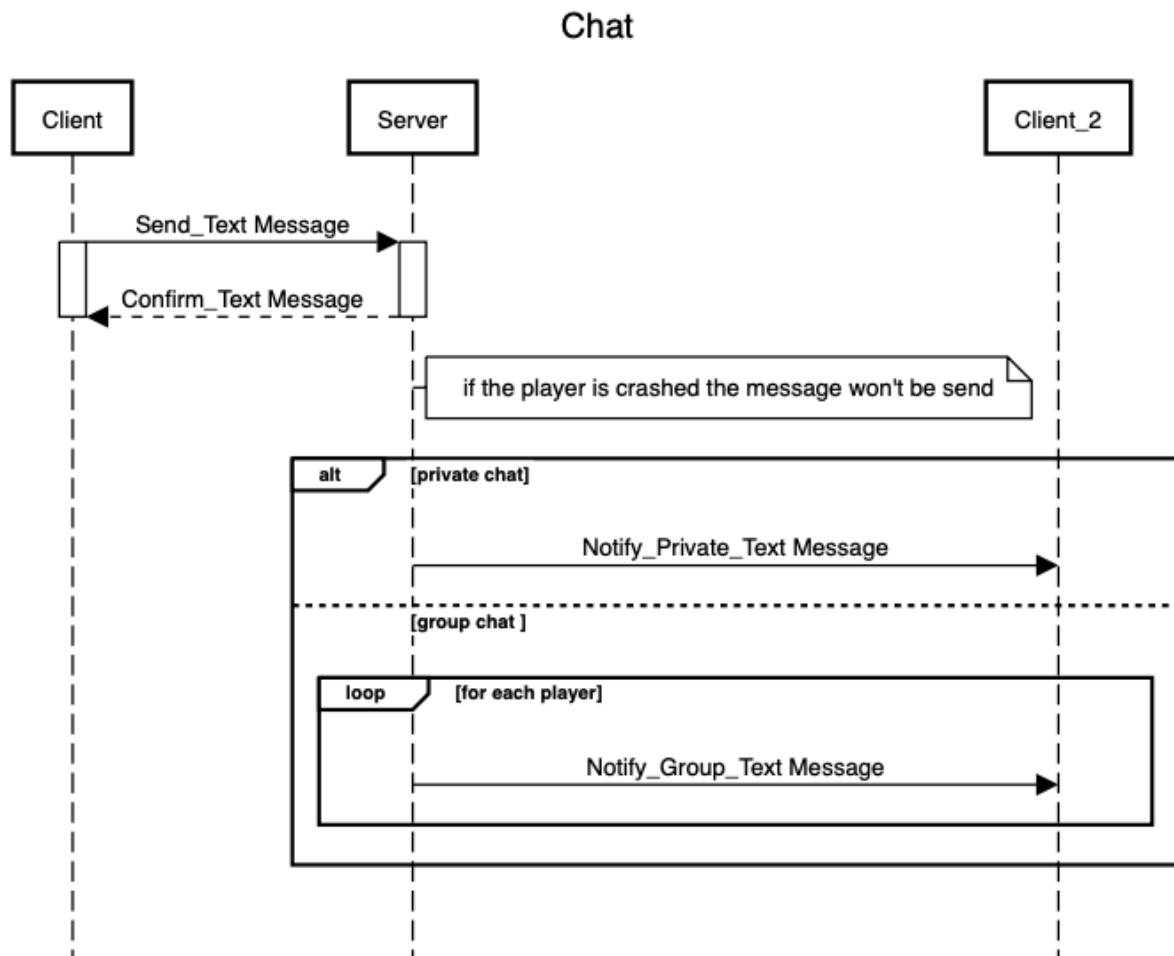
- `NotifyOnActivePlayerMessage(String message)`.
- `PlaceCardMessage(String nickname, int choice, int i, int j, boolean face)`:
The method called in Game Controller is `PlaceCard`, which calls on Game Model the method `PlaceCard`. The request will be denied if:
 - 'nickname' isn't the current player
 - Game state isn't "PLAYING"
 - 'choice' <1 or >3
- `ConfirmPlaceCardMessage(Boolean confirm, String errorDetails)`.
- `UpdatePlayerBoardMessage(String player, int i, int j, PlayableCard p)`.
- `DrawTableMessage(String nickname, int choice)` : with this message a client can draw from one of the four cards on the common board. The method called in Game Controller is `DrawTable`, which calls the homonymous method in the model. If the game state is not "DRAWING" or the player making the request is not the current player or the selected card does not exist the request will be denied.
- `ConfirmDrawTableMessage(Boolean confirm, String errorDetails)`.
- `DrawResourceDeckMessage(String nickname)`: with this message a client can draw from the resource deck. The chosen deck is empty.
The method called in Game Controller is `DrawResourceDeck` which calls the homonymous method in the model. If the game state is not "DRAWING" or the player making the request is not the current player or the deck is empty.
- `ConfirmDrawResourceMessage(Boolean confirm, String errorDetails)`.
- `DrawGoldDeckMessage(String nickname)` with this message the client can draw from the gold deck. The method called in Game Controller is `DrawGoldDeck` which calls the homonymous method in the model. If the game state is not "DRAWING" or the player making the request is not the current player or the deck is empty.
- `Confirm Draw Gold Message (Boolean confirm, String errorDetails)`.
- `UpdatePersonalCardsMessage(String player, ArrayList <PlayableCards> a)`.
- `UpdatePlayerPoints Message (String player, int updated, int delta)`.
- `UpdateCommonBoard Message(ResourceDeck r, GoldDeck g, ArrayList<PlayableCards> p)`.

End game



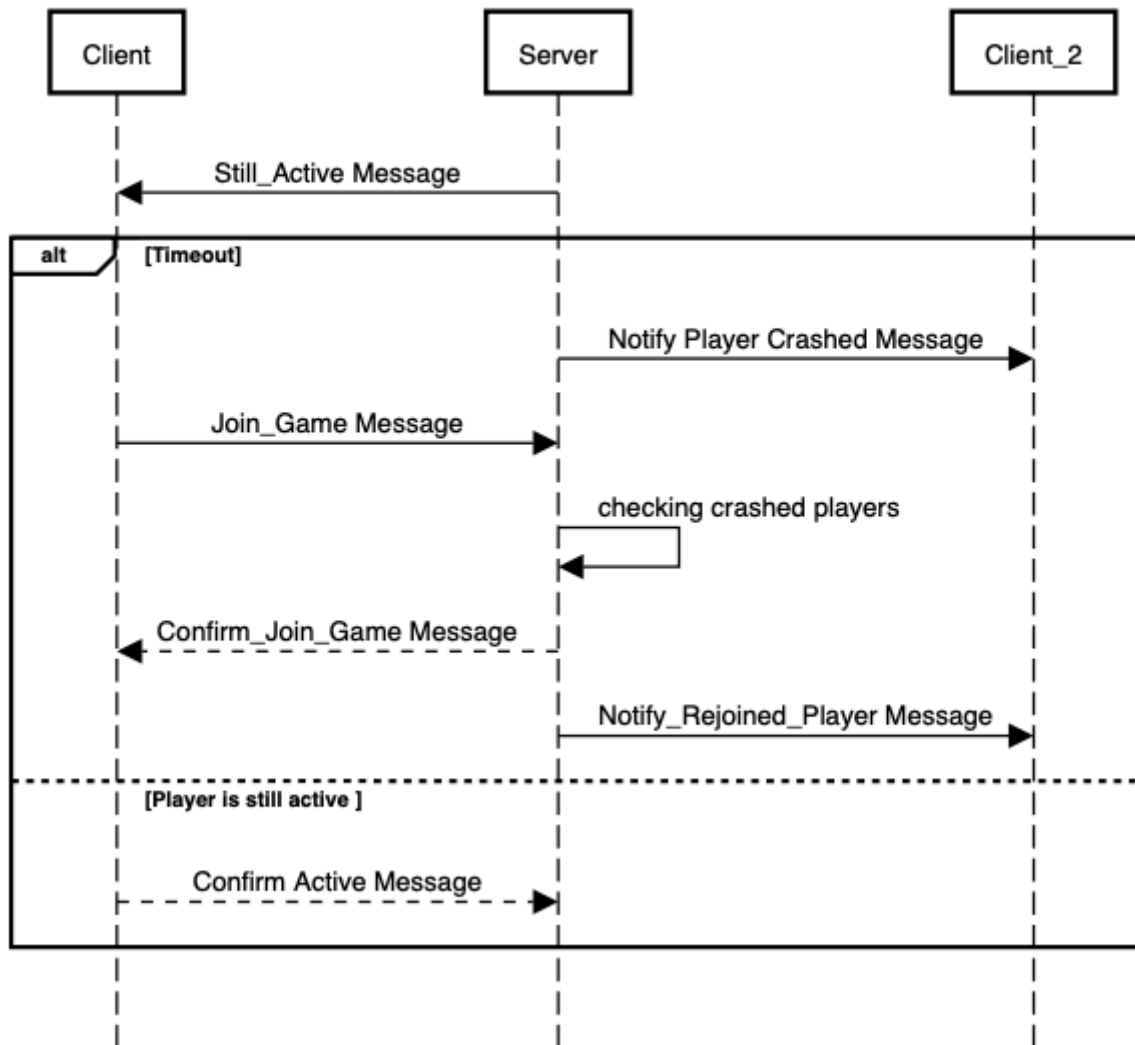
- NotifyAdditionalRoundMessage (String message).
- NotifyEndingGame Message (String message).
- NotifyWinnersMessage (ArrayList<Player> p, String message).

Advanced Features



- `SendMessage` (String sender, String rec, String content):
 - The request will be denied if the recipient is a crashed player.
- `ConfirmTextMessage` (Boolean confirm, String errorDetails)
- `NotifyPrivateTextMessage` (String Sender, String rec, String content)
- `NotifyGroupTextMessage` (String Sender, String content)

Resilience To Disconnections



- StillActiveMessage
- ConfirmActiveMessage
- NotifyPlayerCrashed (String message)
- RejoinGameMessage (String nickname):
 - The request will be denied if there is a game with at least one crashed player inside, or if the nickname does not match the one previously used.
- ConfirmJoinGameMessage (Boolean confirm, String errorDetails)
- NotifyRejoinedPlayer (String message)