

Network Protocol

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

Gruppo AM03

This document presents the network protocol between client and server.

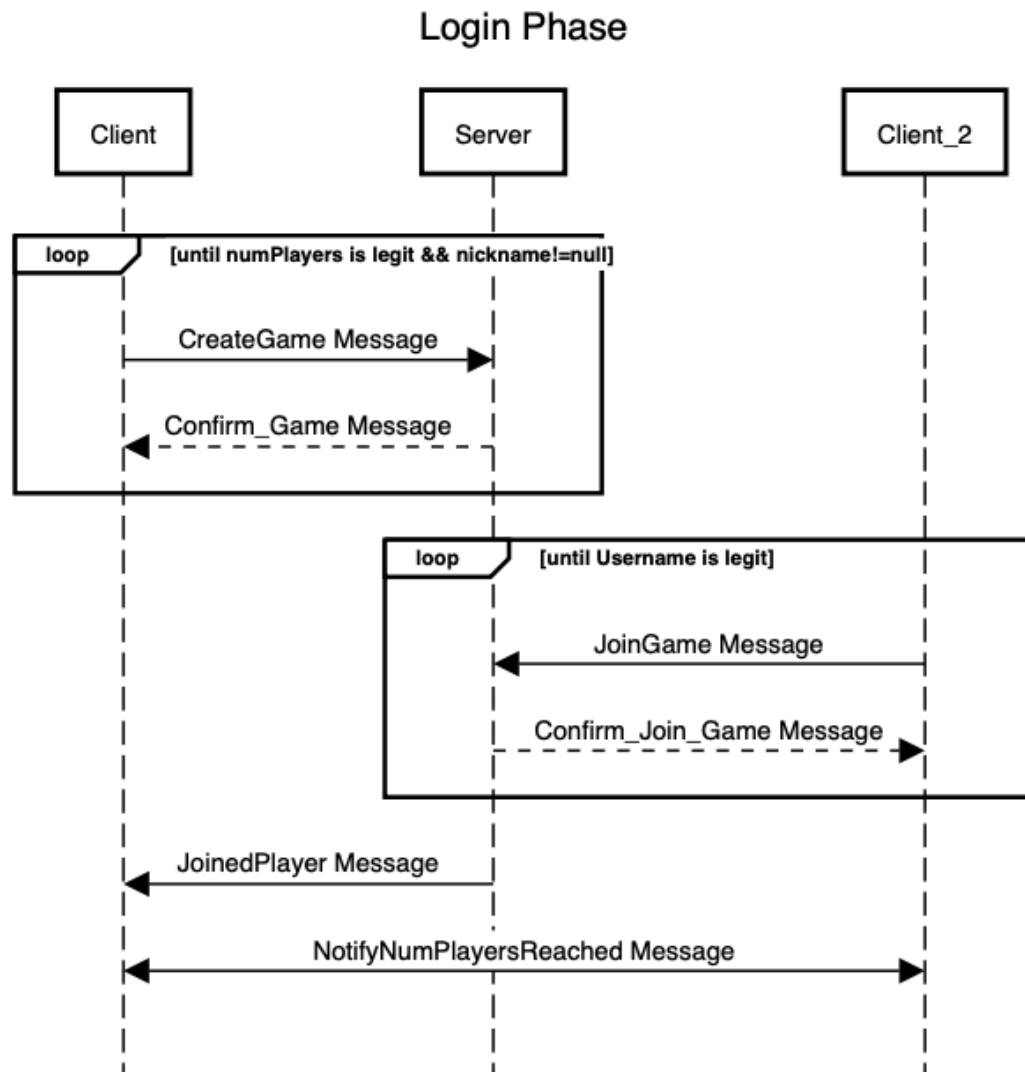
Subscribers note

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

- Player Subscriber
- Game Subscriber
- PlayerBoard Subscriber
- Chat Subscriber

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

Login

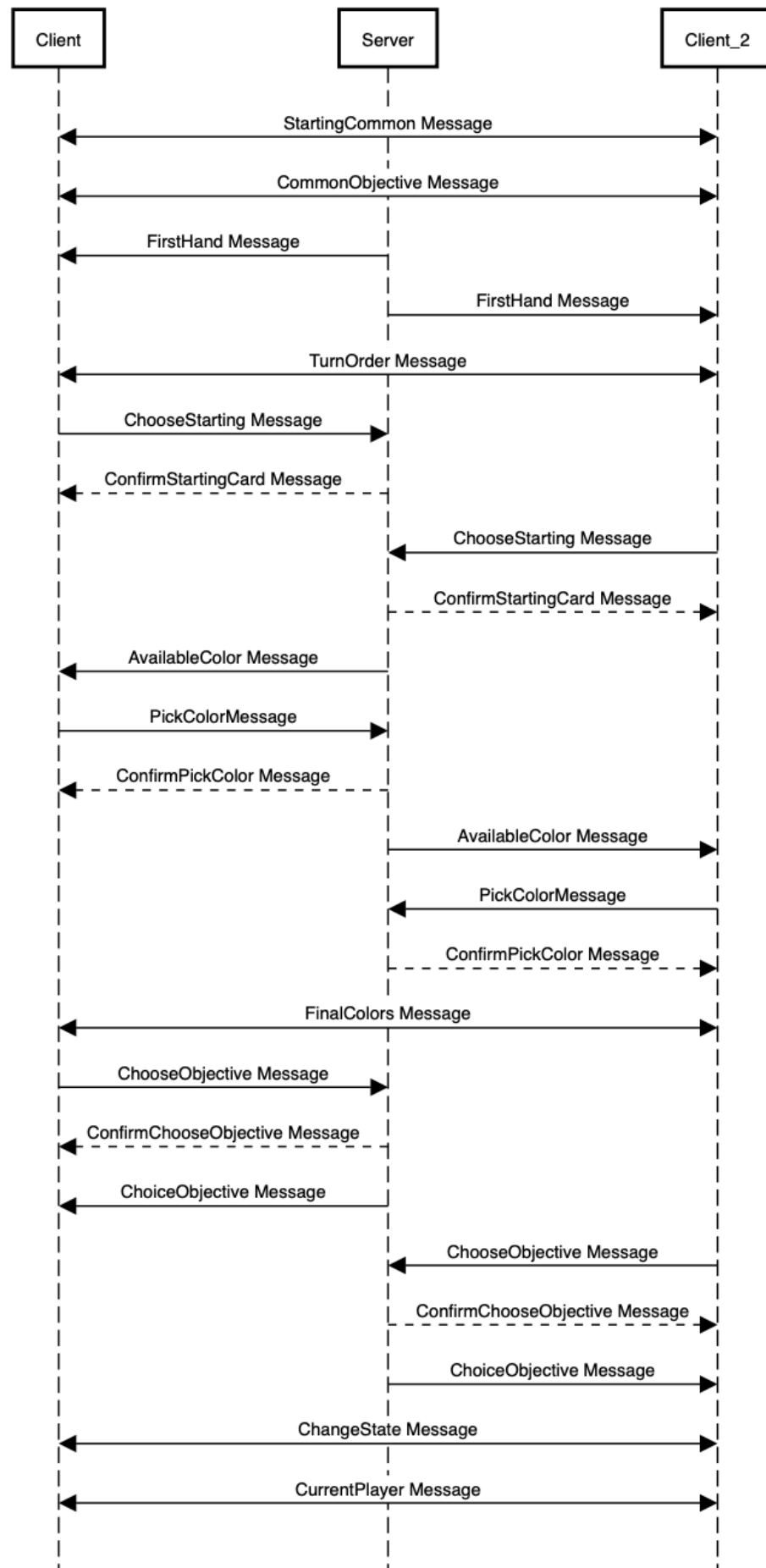


- CreateGameMessage (int numPlayers, String nickname):
The game controller calls createGame(int numPlayers, String nickname) which creates an instance of Game Model. The request is denied if :
 - *numPlayers* is <2 or >4
 - *nickname* is a blank space
- ConfirmGameMessage (Boolean confirmGameCreation, String details, String nickname).
- JoinGameMessage (String nickname, Boolean hasJoined). The request is denied if:
 - the game has already started
 - the nickname is already taken

- ConfirmJoinGameMessage(Boolean confirmJoinGame, String details, String nickname).
- JoinedPlayerMessage (String player):
Notification to every player that a new player having as a nickname *player* has joined the game.
- NotifyNumPlayersReachedMessage ():
Notification to every player that the number of players has been reached, so the game can start.

Game Set-Up

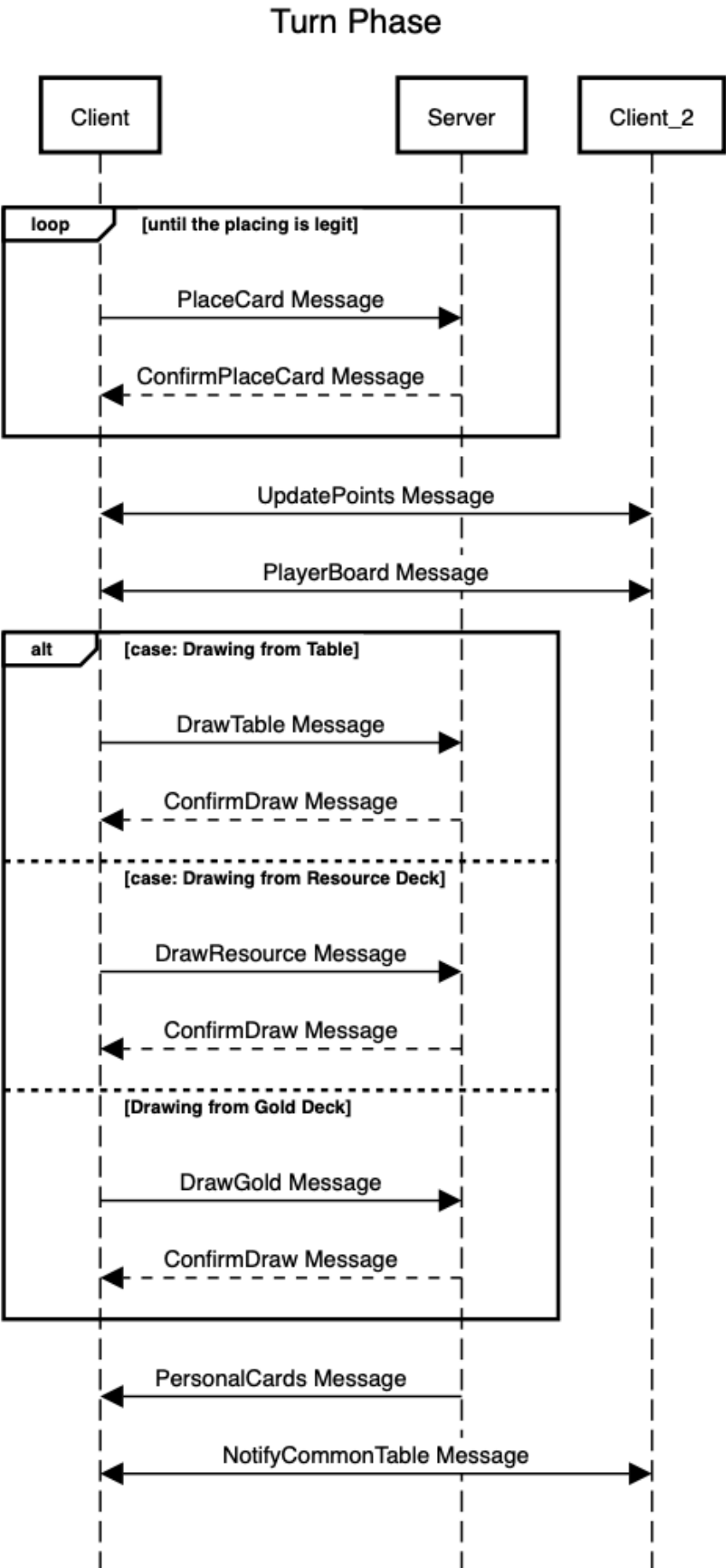
Game Set-Up phase



- StartingCommonMessage (ArrayList<ResourceCard> commons):
Notification to every player about the six cards, contained in *commons*, that can be drawn from the game table .
- CommonObjectiveMessage (ObjectiveCard o1, ObjectiveCard o2):
Notification to every player about the two common objective cards, respectively *o1* and *o2*.
- FirstHandMessage (ResourceCard playablecard1, ResourceCard playablecard2, ResourceCard playablecard3, StartingCard startingCard, ObjectiveCard o1, ObjectiveCard o2):
Notification to every player about their first hand's cards, starting card and the two objective cards from which he will have to choose.
- TurnOrderMessage (ArrayList<String> turnOrder):
Notification to every player about their turn order.
'turnOrder' contains nicknames of players ordered according to their turn order
- ChooseStartingMessage (String player, String face):
The method called in Game Controller is selectStartingFace, which calls on Game Model the method .canSelectStartingFace(player, face). If the player isn't the valid one or the Game State isn't "STARTING" are invalid the request will be denied. *face* needs to be "FRONT" or "BACK".
- ConfirmStartingCardMessage (Boolean confirmStarting, String details)
- AvailableColorMessage (ArrayList<Color> colors):
This message notifies the player who has to pick the pawn's color about the available colors which can be chosen.
- PickColorMessage (String nickname, String color)
The game controller calls canPickColor(String player, String color) to verify if the request can be satisfied. The request is denied if :
 - *player* is not the valid one
 - *color* is not available or doesn't exist
- ConfirmPickColorMessage (Boolean confirmPickColor, String details)

- FinalColorsMessage (Map<String, Color> colors):
Notification to every player about the color picked by each player.
- ChooseObjectiveMessage (String player, int choose):
The method called in Game Controller is canSetObjectiveCard(String player, int choice) which calls on Game Model the method .setObjectiveCard(nickname, choice). If the player isn't the valid one or the Game State isn't "OBJECTIVE" the request will be denied. *choose* needs to be '1' or '2'.
- ConfirmChooseObjectiveMessage (Boolean confirmChoice, String details).
- ChoiceObjectiveMessage (String player, ObjectiveCard objectiveCard):
Notifies the player having as a nickname *player* with the personal objective card that he has just selected, allowing him to update his local model.
- ChangeStateMessage (State state):
Notifies every player that the game state has changed.
- CurrentPlayerMessage (String currentPlayer):
Notifies every player about the new current player having as a nickname *currentPlayer*

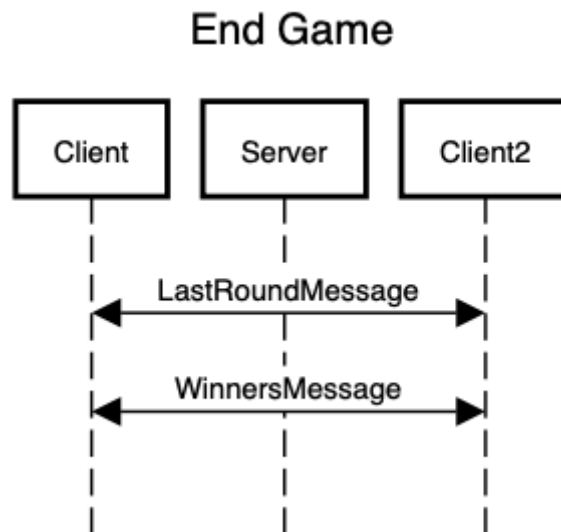
Turn Phase



- PlaceCardMessage(String player, int choice, int i, int j, String face): The method called in Game Controller is canPlaceCard, which calls on Game Model the method PlaceCard. The request will be denied if:
 - *player* isn't the current player
 - Game state isn't "PLAYING"
 - *choice* <1 or >3
 - indexes are not valid
 - face is not valid
- ConfirmPlaceCardMessage (Boolean confirmPlacing, String details).
- UpdatePointsMessage (String player, int points):
Notification to every player about the point *points* earned by the player *player*.
- PlayerBoardMessage(String player, PlayableCard playableCard, int i, int j):
Notification to every player about the player board updated of the player *player* after the card *playableCard* has been placed on his board.
- 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 canDrawTable, which calls the drawTable method in the model. If the game state is not "DRAWING" or the player having as a nickname *nickname* making the request is not the valid one the request will be denied. *choice* needs to be > 0 and < 5.
- ConfirmDrawMessage(Boolean confirmdraw, String details).
- DrawResourceMessage(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 canDrawResource which calls the drawResources method in the model. If the game state is not "DRAWING" or the player having as a nickname *nickname* making the request is not the valid one the request will be denied. The resource deck needs to contain at least one resource card.

- ConfirmDrawMessage(Boolean confirmdraw, String details).
- DrawGoldMessage(String nickname):
 With this message the client can draw from the gold deck. The method called in Game Controller is canDrawGold which calls the drawGold method in the model. If the game state is not "DRAWING" or the player having as a nickname '*nickname*' making the request is not the valid one the request will be denied. The gold deck needs to contain at least one gold card.
- ConfirmDrawMessage(Boolean confirmdraw, String details).
- PersonalCardsMessage(String player, ArrayList <ResourceCard> hand):
 Notifies the player having as a nickname *player*, which has just drawn a card, about his new personal cards contained in *hand*.
- NotifyCommonTableMessage(ResourceCard resourceCard, int index):
 Notifies all players of a new card *resourceCard* on the game table that has replaced one that was drawn. *index* is used to identify which one of the six cards on the game table has changed.

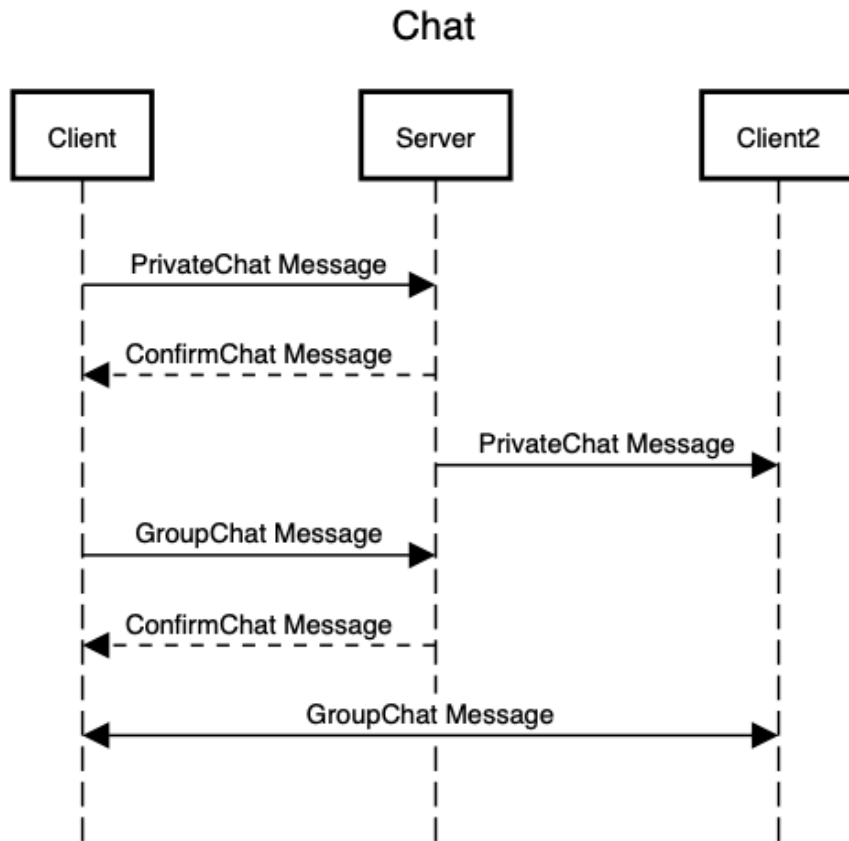
End game



- LastRoundMessage ():
Notification to every player that the last round has started.
- WinnersMessage (ArrayList<Player> winners):
Notifies all players about the winners.

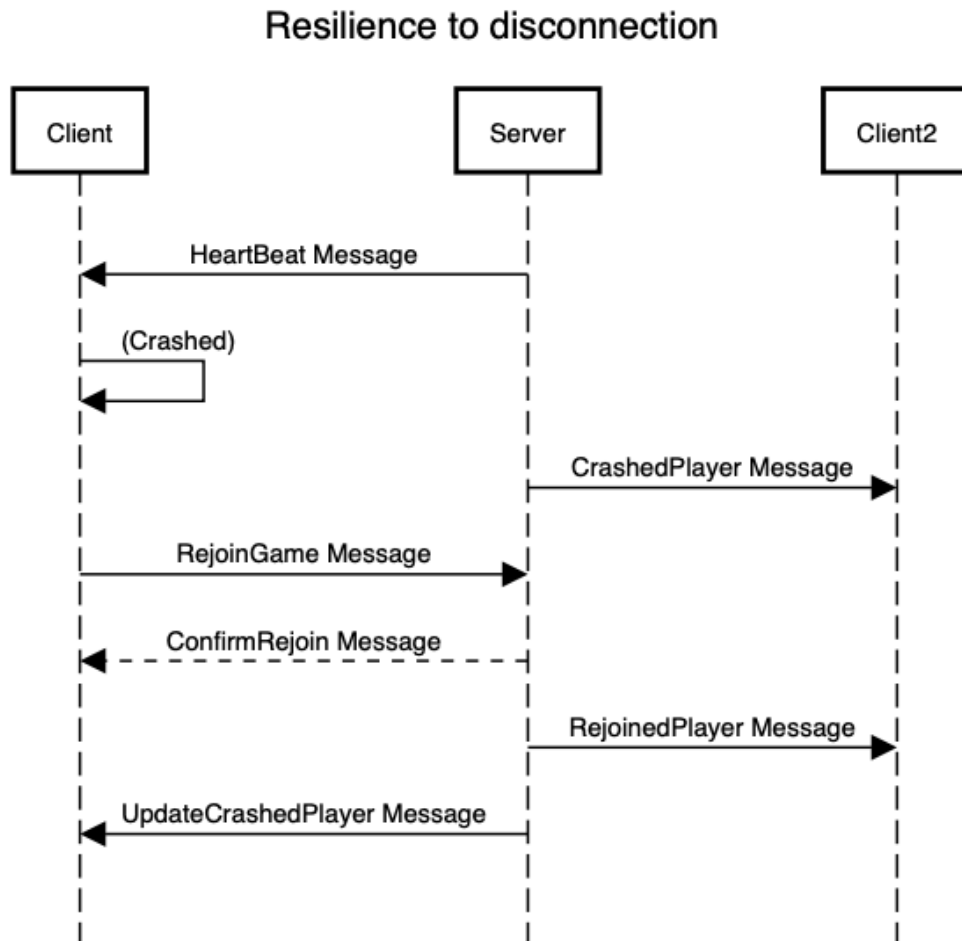
Advanced Features

Chat



- PrivateChatMessage (String sender, String recipient, String text):
 - The request will be denied if the recipient isn't in the game (crashed player/not existing player)
- GroupChatMessage (String sender, String text)
- ConfirmChatMessage (Boolean confirmChat, String details)

Resilience to disconnection



- HeartbeatMessage (String heartbeat):
message used to check the connection between client and server.
- CrashedPlayerMessage (String player):
Notification to every player that the player having as a nickname *player* has crashed.
- RejoinGameMessage (String nickname):
The request will be denied if the game has already ended or if the nickname is not of a player crashed.
- ConfirmRejoinGameMessage (Boolean confirmRejoinGame, String details)
- RejoinedPlayerMessage (String player):
Notification to every player that the player having as a nickname *player* has rejoined
- UpdateCrashedPlayerMessage (String nickname, ArrayList<Text> chat, State gameState, ArrayList<ResourceCard> hand, ObjectiveCard objectiveCard, Map<String, PlayableCard[][]> boards, Map<String, Integer> points, ArrayList<String> players, ArrayList<ObjectiveCard> objectiveCards, Color

color, ArrayList<ResourceCard> table, ArrayList<Color> colors):

This message is used to send all the information about the current state of the game to the player *nickname* who has just rejoined.