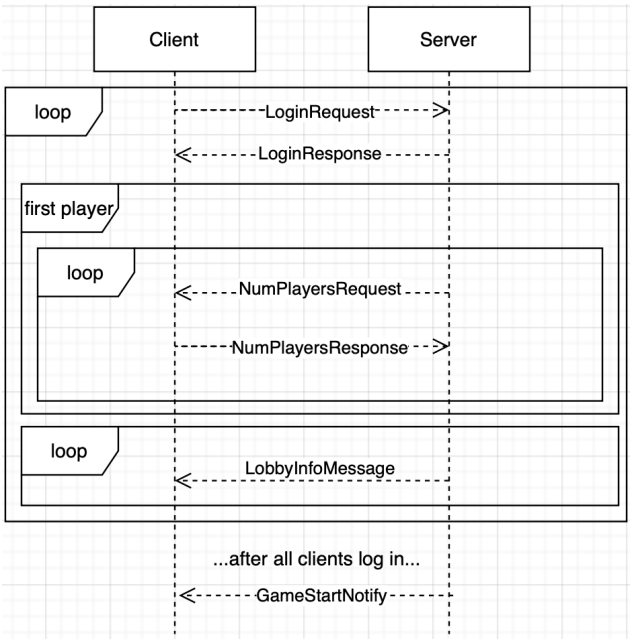
Communication Protocol

Simone Bevilacqua 10720775 Raffaele Chiaverini 10711746
Luca Barda 10697790 Michael Alibeaj 10750827
Gruppo AM-11

The phases of the communication between client and server are the following.

Connection phase

When the client connects to the server (either using TCP or RMI protocol) it sends a ***LoginRequest*** message to the server containing the nickname to use during the game. The server checks if the nickname is already taken and sends back a ***LoginResponse*** message containing the nickname (if valid). This will keep going until the client sends a valid nickname.
When the client provides a valid username, the server saves the information about the new client and checks if it's the first player joining the lobby, in that case, it sends a ***NumPlayersRequest*** message and waits for a ***NumPlayersResponse*** message from the client containing the number of players that will play the game. This will keep going until the client sends a valid number of players.
When a new player connects/disconnects to/from the game, the server notifies all players currently waiting in the lobby with a **LobbyInfoMessage** message containing the name of the player involved, a flag to determine whether it's a connection or a disconnection and the new lobby.



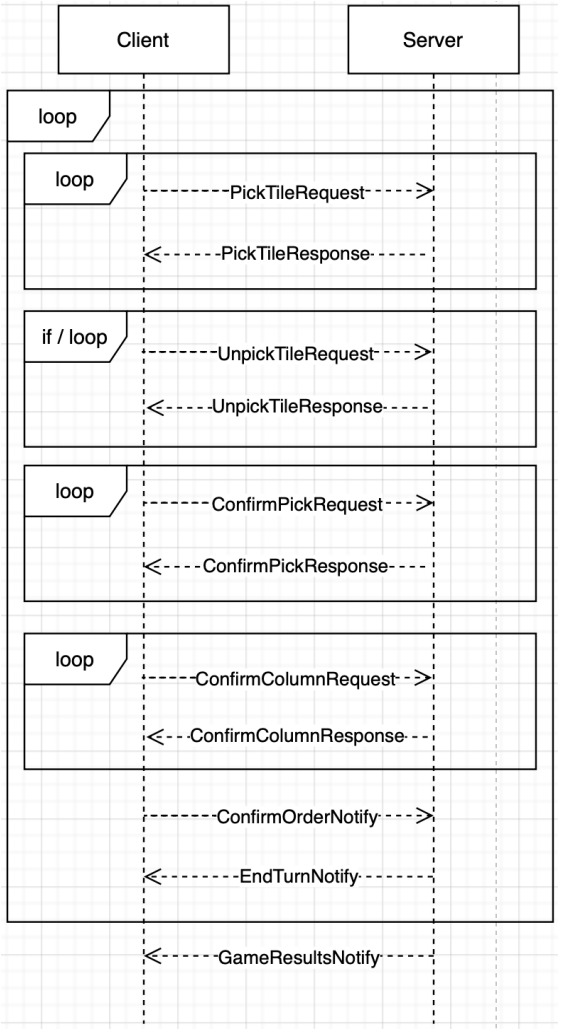
Game initialization phase

When the required number of players have connected to the game, the server sends to each client the required items to play the game inside a ***GameStartNotify*** message which contains:

- a ***Board*** object: contains a set of coordinates with the corresponding tiles;
- two ***CommonGoal*** objects: each contains a common goal picked for the game;
- two **Token** objects on top of each common goal;
- a ***PersonalGoal*** object: contains the personal goal assigned to the client;
- a list of the players' nicknames;
- the first player's nickname.

Game phase

The server waits for the clients' moves.
The client sends a ***PickTileRequest*** message specifying the coordinates of the tile to pick. The server checks if it's a valid pick and sends back a ***PickTileResponse*** message containing either true or false.
The client can also send a ***UnpickTileRequest*** message specifying the coordinates of the picked tile that has to be removed from the picked ones. The server checks if it's a valid tile to remove and sends back a ***UnpickTileResponse*** message containing either true or false.
The client sends a ***ConfirmPickRequest*** message to confirm the picked tiles. The server sends back a ***ConfirmPickResponse*** message containing either true or false and, in the first case, the updated Board.
The client sends a ***ConfirmColumnRequest*** message to choose the column where to insert the picked tiles and waits for a positive ***ConfirmColumnResponse*** message.
In the end, the client sends a ***ConfirmOrderNotify*** message containing the final order of the picked tiles.



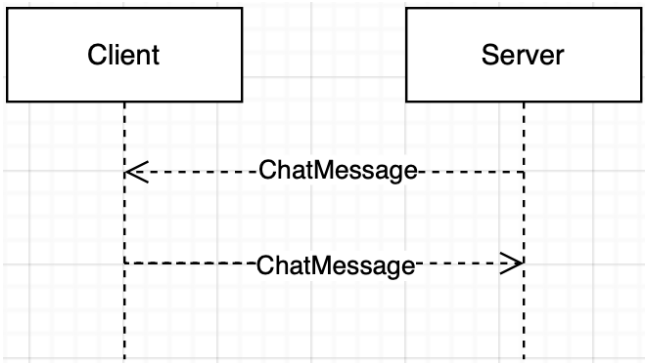
At this point, the server sends to every player an ***EndTurnNotify*** message containing the new bookshelf and the new points of the last player plus the next player's nickname.
During the game it is possible that a player disconnects from the game, in this case the server sends to all the other players a **PlayerDisconnectedNotify** message containing the name of the disconnected player.

End Game phase

At the end of the last turn, the server sends a ***GameResultsNotify*** message to every player containing each player's points and the nickname of the winner. At this point, the server closes the connection with all the clients and the game ends.

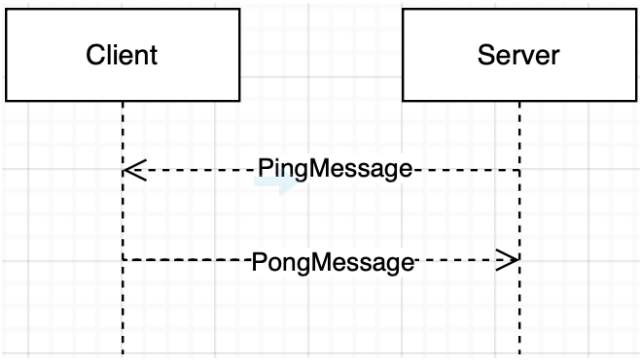
Chat

During the game each client can send a ***ChatMessage*** message specifying the text of the message, its nickname and the nickname of the player to send the message to. If the receiver's nickname is not specified the message is sent to everyone. The server sends the ***ChatMessage*** message either to the selected player or to every player.



PingPong

During the game the server keeps testing the connection with each client. The server sends a ***PingMessage*** message to the client, then waits for a response (***PongMessage*** message). If the server doesn't receive a response within 20 seconds, the server considers the last message as lost.
When the server sends 5 consecutive messages without any response to the same client, it considers the connection with that client closed and ends the game.



Messages

The messages used during the communication are:

- LoginRequest {
  String nickname
  }
- LoginResponse {
  String nickname
  }
- NumPlayersRequest {}
- NumPlayersResponse {
  int numPlayers
  }
- LobbyInfoMessage {
  int sizeLobby,
  List<String> lobby,
  boolean newPlayerConnected,
  String playerName
  }
- GameStartNotify {
  Board board,
  List<CommonGoal> commonGoals,
  List<Token> commonGoalsTokens,
  PersonalGoal personalGoal,
  List<String> players,
  String nextPlayer
  }
- PickTileRequest {
  Position position

}

- PickTileResponse {
  boolean valid,
  List<Tile> pickedTiles
  }

- UnpickTileRequest {
  Position position
  }
- UnpickTileResponse {
  List<Tile> pickedTiles,
  boolean successful
  }
- ConfirmPickRequest {}
- ConfirmPickResponse {
  Board board,
  boolean valid
  }
- ConfirmColumnRequest {
  int columnNumber
  }
- ConfirmColumnResponse {
  boolean valid,
  boolean needForSwap
  }
- ConfirmOrderNotify {
  List<Tile> pickedTiles
  }
- EndTurnNotify {
  Board board,
  Bookshelf bookshelf,
  List<Token> commonGoalsTokens,
  int points,
  boolean endGame,
  String player,
  String nextPlayer

}

- GameResultNotify {}
- ChatMessage {
  String player,
  String receiver,
  String textMessage
  }
- PingMessage {}
- PongMessage {}