Explanation of the protocol

The protocol has 3 phases:

- entering lobby phase
- waiting in lobby phase
- game phase

Every kind of message causes a method call on the machine on the other side of the connection

RMI connection

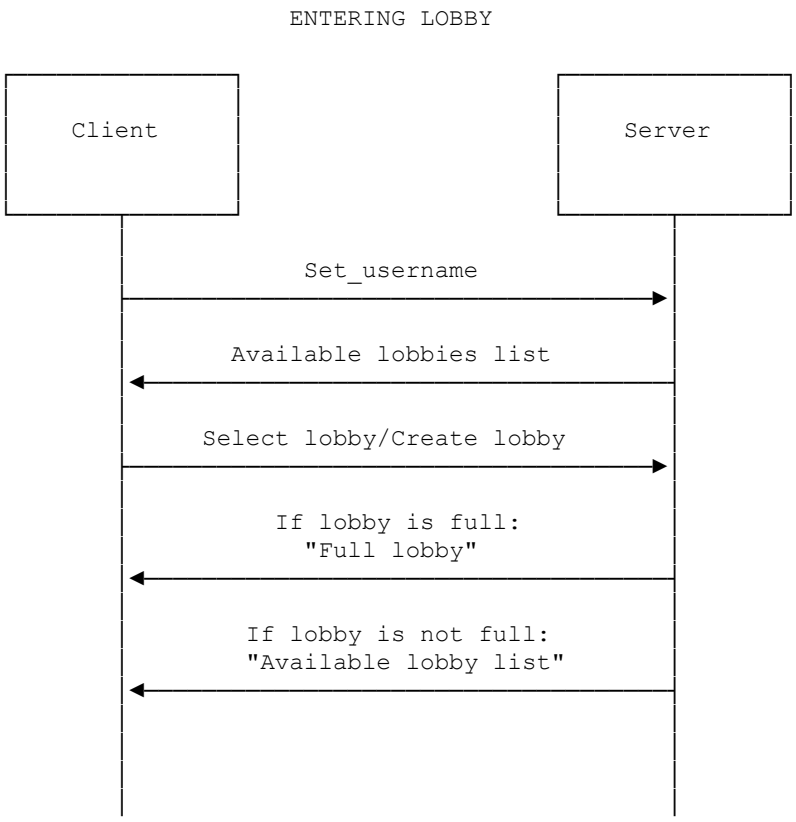
In the case of an RMI connection, every message is actually a direct method call on the View or Controller objects.\

TCP connection

In the case of a TCP connection, every possible type of message is represented by a class,
which has private attributes representing the arguments of the method called on the View/Controller, the sender creates a new object representing the message, and uses the Serializable java interface to send it through the connection,
the receiver then deserializes the received object, and calls the run() method on the message object,
which then calls the corresponding method on the receiving View/Controller object.

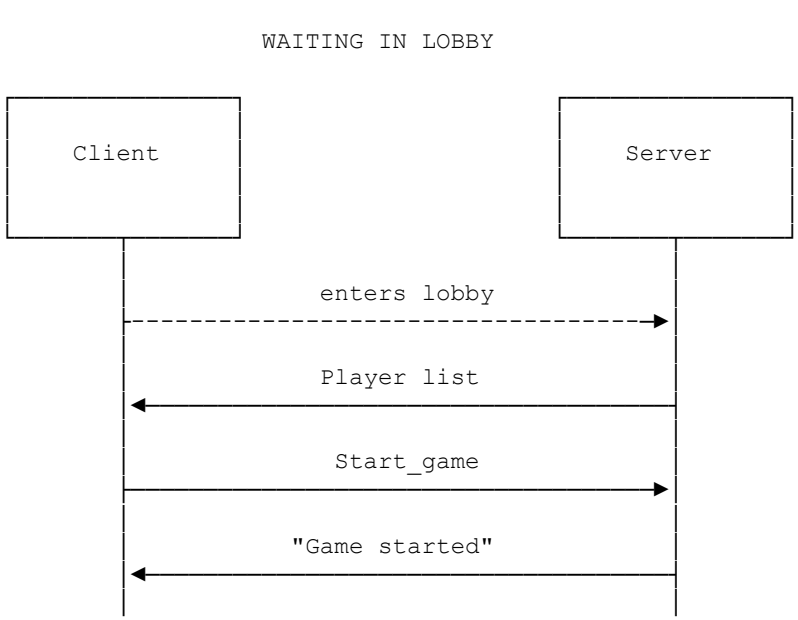
Entering lobby phase

In this phase, the user chooses his username,
then chooses to either join a lobby, or to create a new one



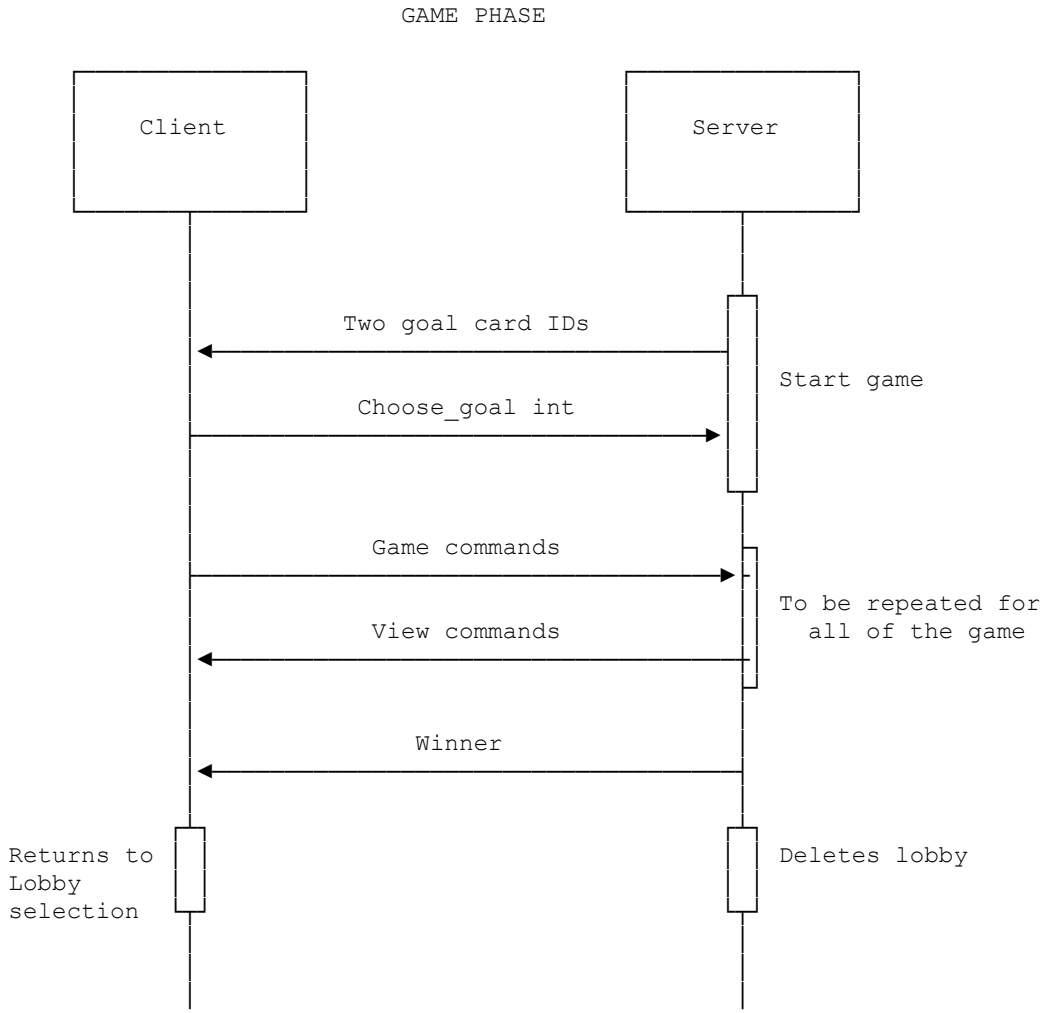
Waiting in lobby phase

In this phase, every player can choose to leave the lobby,
but only the first player (called admin) can start the game,
if there are at least 2 players.



game phase

in this phase messages are exchanged between client and server to play the actual game.
the client sends **game commands** to describe the player's actions on the game.
the server sends **view commands** to update the client's view to reflect the game model after a change.



Command List

Lobby commands:

set_username (String) "username": sets the username
select_lobby (int) lobby_number: chooses which lobby to join

create_lobby: creates a new lobby and joins it
quit_lobby: quits the lobby and returns to lobby selection
start_game: starts the game (succedes only if the player is the admin and if there are 2 or more players)

Game commands:

da notare che ogni game command sarà preceduto dallo username del giocatore che l'ha invocato (String) e il numero di lobby in cui partecipa (int)

choose_goal (int) index: when presented the 2 goal cards to choose from, it chooses one of them (0 for left, 1 for right)
draw_face_up_gold (int) index: draws a card from the two gold cards that are facing up
draw_face_up_resource (int) index: draws a card from the two resource cards that are facing up
draw_deck_gold (int) index: draws from the gold card deck
draw_deck_resource (int) index: draws from the resource card deck
place_card (int) index (int) x (int) y: places a card from the player's hand, identified by its index (0,1,2), and places it on their table
message (String) receiver (String) "message": adds a message to the chat, (the receiver is null for global messages)

View updates:

announce_unexpected_disconnection (String) username: this command is sent when a player disconnects from the game, resulting in the game ending prematurely
player_number (int) num: tells the player which turn number he has
present_goal_card_options (int) id1 (int) id2: presents the player 2 goal cards to choose from
update_game_state (GameState) state: updates the game state
update_round (int) round: updates the round number
update_turn_player (String) username: updates the player who is currently playing
add_player (String) username: adds a player to the lobby
remove_player (String) username: removes a player to the lobby
update_upside_up_resource (int) id: updates which upside up resource card is on the table when a player draws one
update_upside_up_gold (int) id: updates which upside up gold card is on the table when a player draws one
update_upside_up_goal (int) id: tells the client the common goal cards at the start of the game
resource_card_deck_ announce_winner (String) username: announces the game is over, and who won
add_message (String) sender (String) receiver (String) message: adds a message to the chat (if the message is global, the receiver is null)
add_card_to_table (String) username (int) id (int) x (int) y: adds a card to the player's table in coordinates (x, y)
update_points (String) username (int) points: updates the player's points
update_symbol_count (String) username (Symbol) symbol (int) count: updates the player's symbol count\