COMMUNICATION PROTOCOL

Nemanja Antonic, René Bwanika, Chiara Buonagurio - AM10

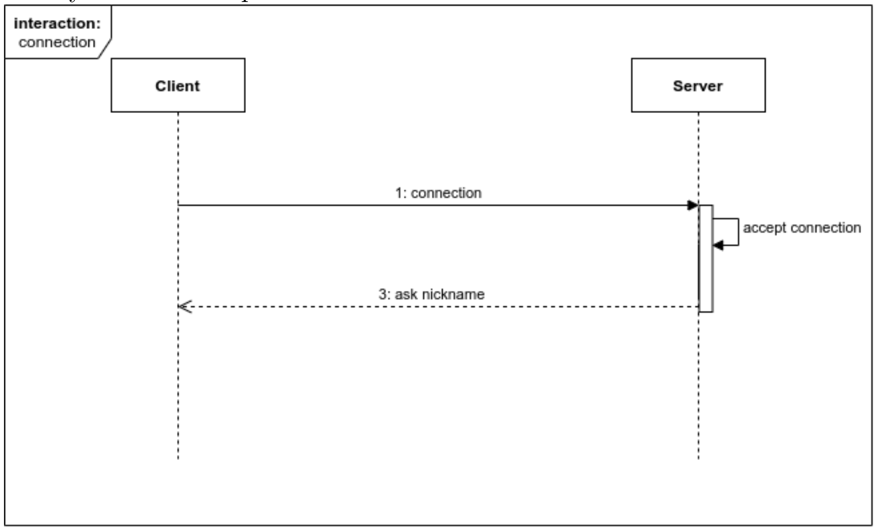
Our communication protocol is based on the serialization and exchange of java objects through socket object streams. We allow the creation of multiple matches by using differents lobbies in which clients connect before starting the game. When a match is created the server is ready to create another. Equal nicknames are forbidden in the same match. There are two phases:

- The setup phase (which occurs every time a new client connects to the server)
- The in-game phase which starts immediately after the client is assigned to a match

SETUP PHASE

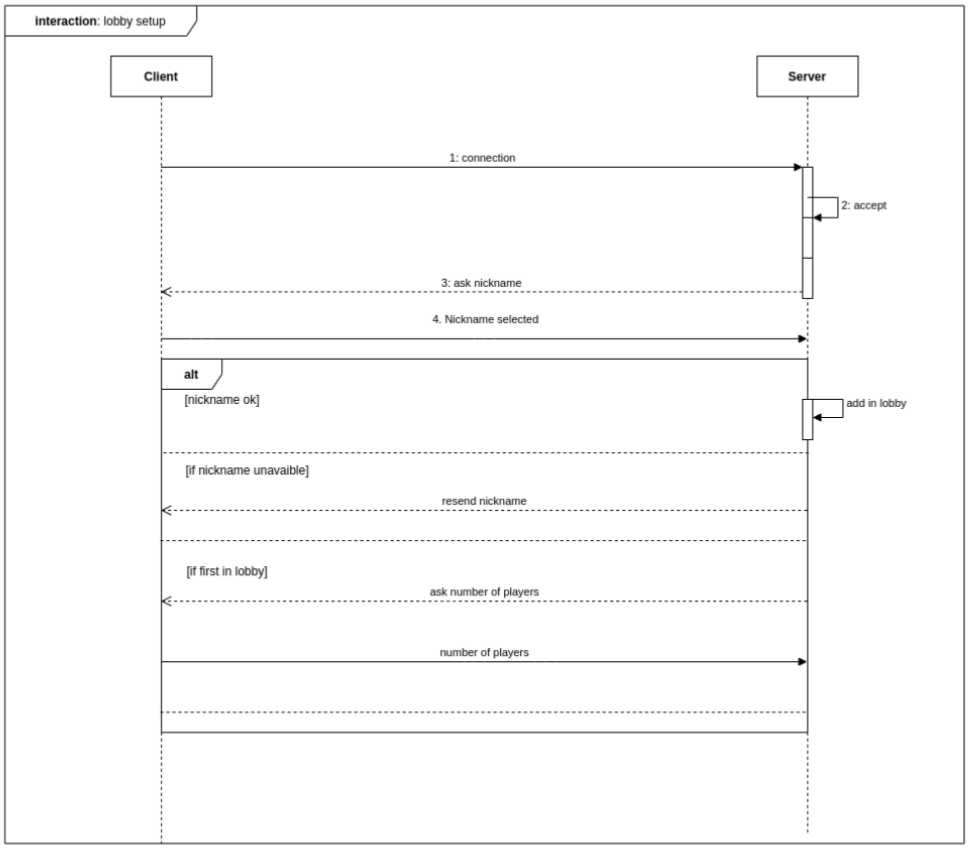
1- connection

The setup phase starts with a client connecting to the server. The client immediately receives a request to insert a nickname.



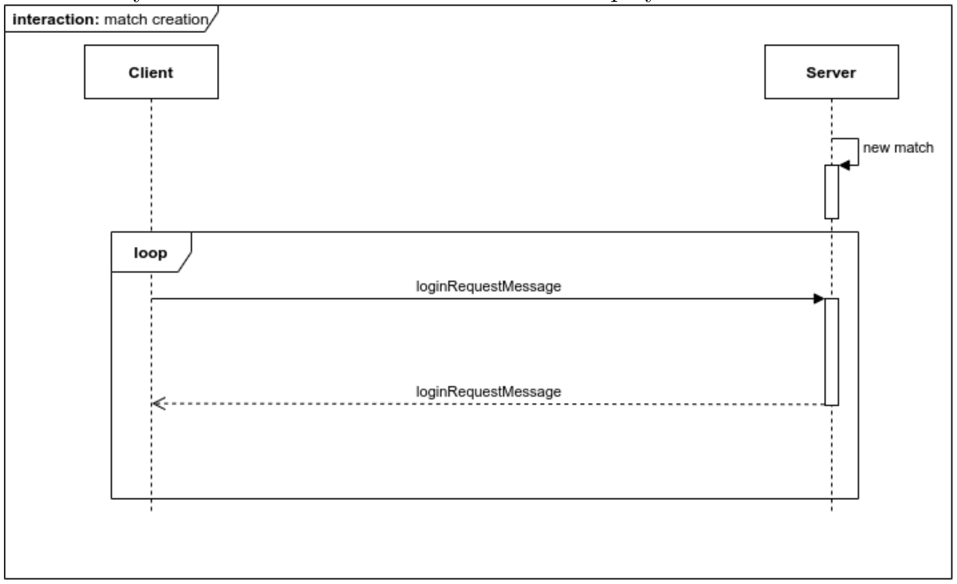
2- setup The server receives a nickname. If the nick name has not been requested by this client or is syntactically invalid the server notifies the client and the flow stops. Otherwise, we have 3 possibilities:

- if the nickname is available, the server add the client to the current lobby
- if the nickname is unavailable, the server asks again a new nickname
- if the client is the first in the lobby, the server asks the number of players for the current match and the server waits for the other clients to join



3- Match creation

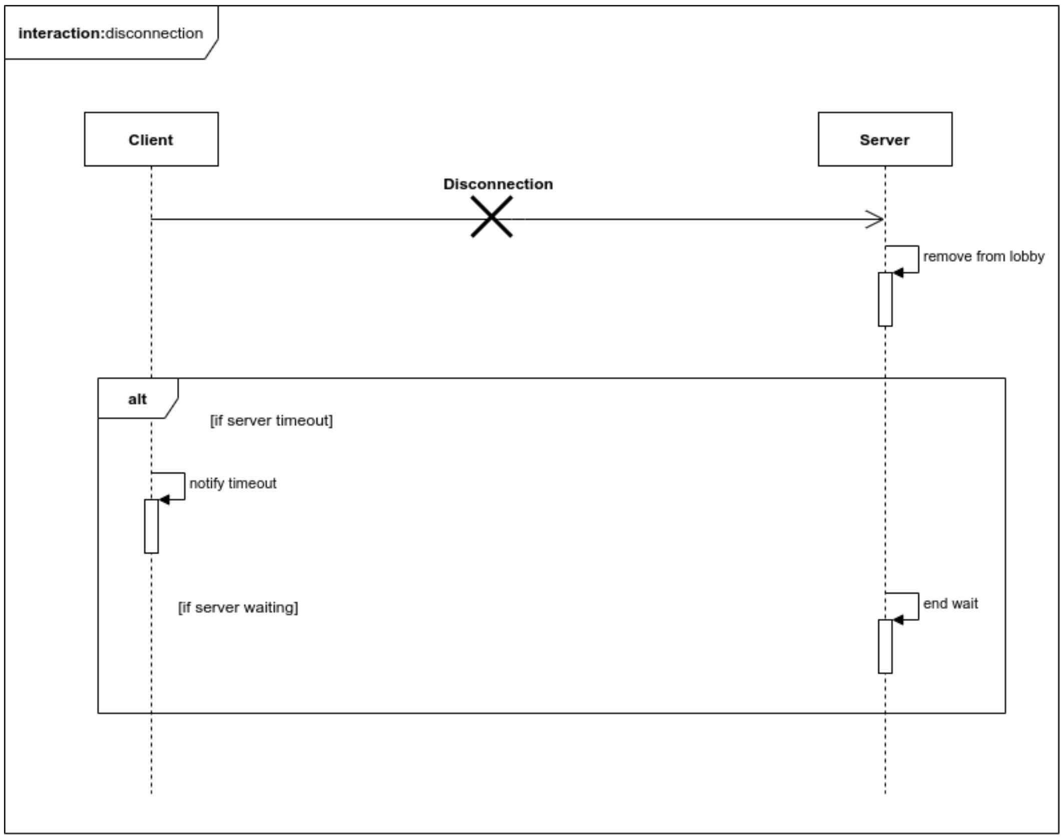
Match creation can occur in many situations when there are enough players to create the match. When a match is created the participating players are notified and are removed from the lobby. If after this, other clients remain in the lobby the server asks the desired number of players to the first in line.



[loginRequestMessage] - [list of players' nicknames, number of players]

4- Disconnection in setup phase

A client is considered disconnected when any type of error occurs when sending or receiving objects. Disconnection can also be forced by the server when a client takes too long to provide a requested packet (this is done by starting an asynchronous timer when sending request packets to the client). When disconnection occurs during the setup phase the client is safely removed from the lobby. If the server was waiting for a packet from this client the wait is resolved. If the disconnection occurs because of the timer the server tries to notify it to the client. Finally the server tries to notify the complete closure of the connection and closes the socket.



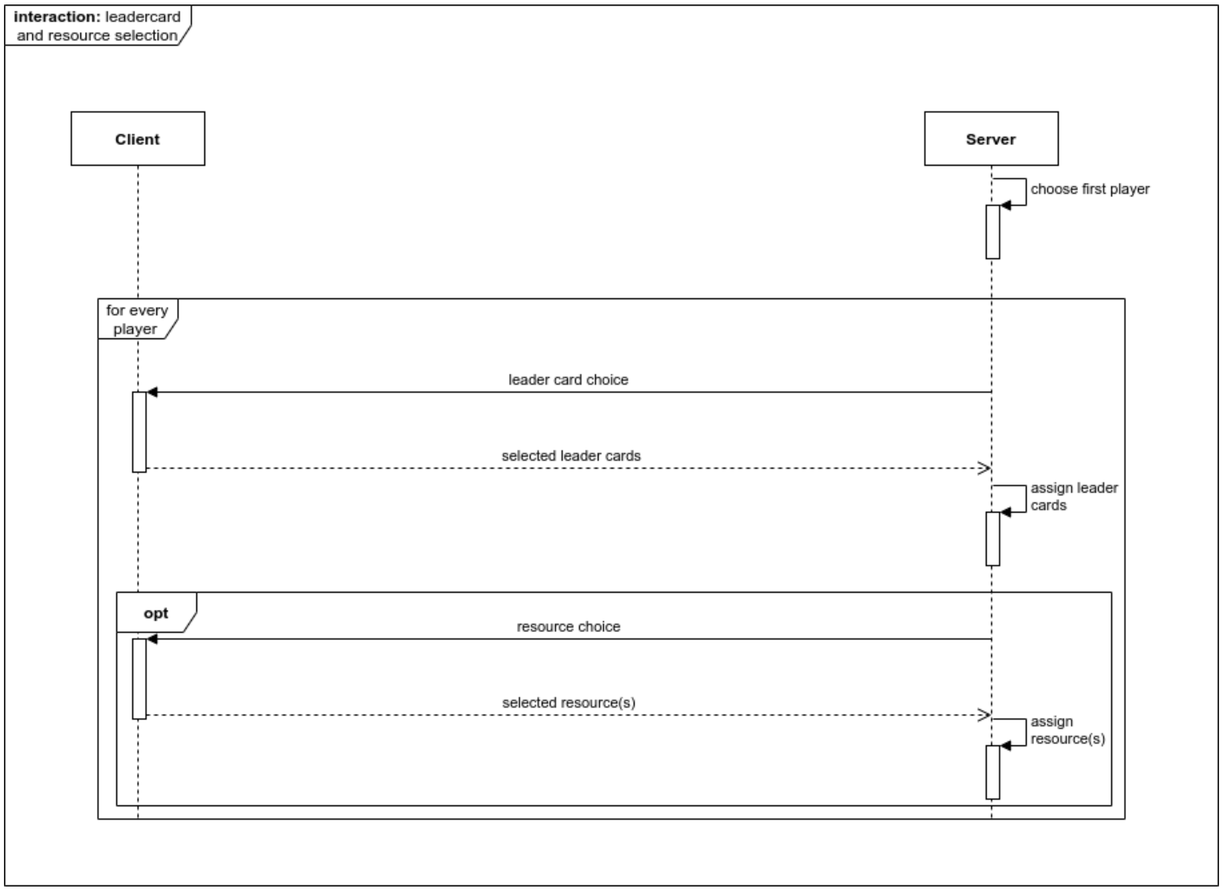
IN GAME PHASE

0: RULES

- A. As a premise, in all the sequence diagrams that will follow we suppose that when a client sends a packet, he has been requested to do so. The cases of non requested incoming packets to the server will result in said packets being ignored by the server itself
- B. If the client provides malformed or not valid information in a packet he has been requested, the server will notify him that the sent packet is invalid and he has to provide a valid packet of the type requested.
- C. Furthermore, when a server requests a packet from the client, a timer is always started; if the client takes too long to provide the requested answer disconnection will be forced by the server.
- D. With each server sent from the server, the view of the clients gets updated to reflect the changes that happened.

1. LEADERCARDS AND STARTING RESOURCES

When a game starts, the server chooses the first player at random. The chosen player then chooses the leadercards. The other players will provide their choices and, eventually, the starting resources.



leader card choice - [ChooseLeadersMessage]

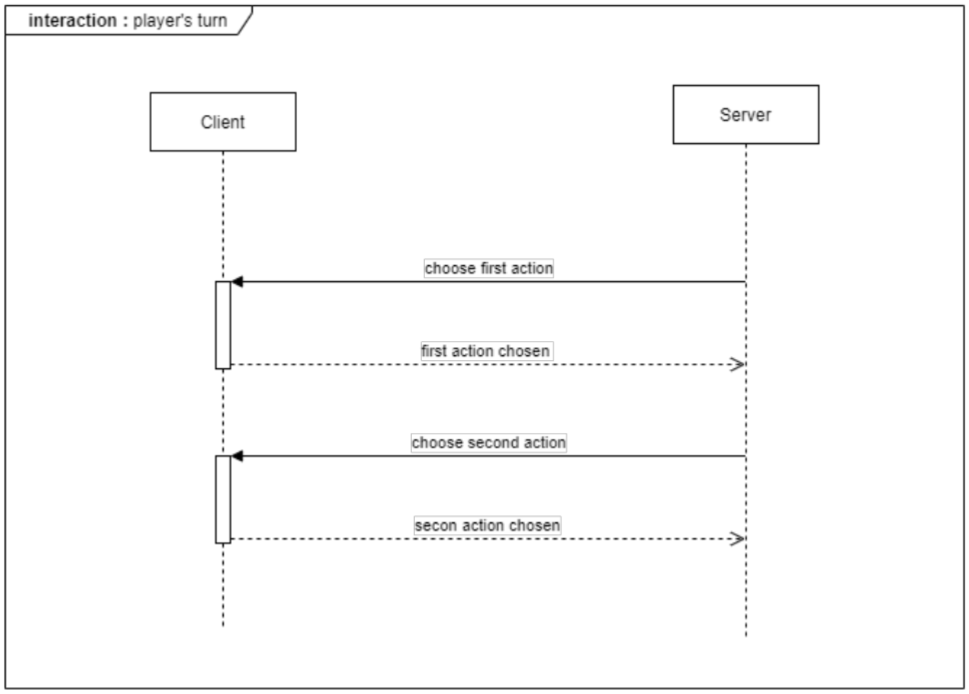
resource choice - [ChooseResourcesMessage]

2. BEGINNING OF PLAYERS TURNS

The turn is divided into a set of actions that the player may perform (see the protocol below). Each action is divided into a set of sub actions, each of which will be a set of options the client can choose from.

PROTOCOL

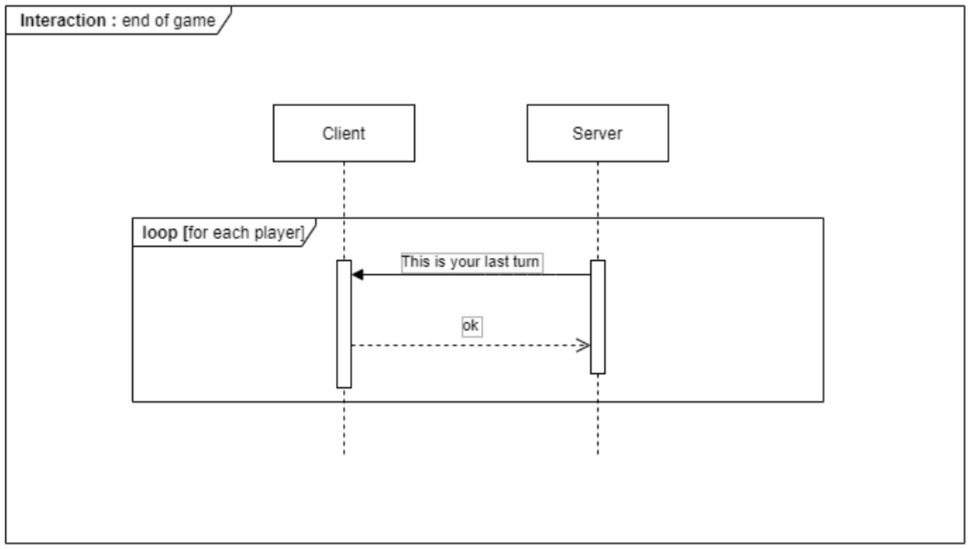
When it is a client's turn, the server asks the client to perform an action. The action can be (buyDevelopmentCard, activateProduction, acquireResources, activateLeaderCard, discardLeaderCard). Once the client is satisfied with his action, he can submit the entire action to the server which will act accordingly. At the end of the turn, the client may perform an action. The action can be (activateLeaderCard, discardLeaderCard, end turn).



3. END OF THE GAME

The moment a player receives their seventh card or reaches the last cell of the Pope road, the server sends all the clients who follow a warning that it is their last turn. When all clients have played their last turn, the server notifies each player of the other players' scores and their score. Then sends the of game finished to all the clients.

Then notifies the connection closure and closes the connection.



6. DISCONNECTION IN GAME A client is considered disconnected when any type of error occurs when sending or receiving objects. Disconnection can also be forced by the server when a client takes too long to provide a requested packet. When a player disconnects from a match, the game keeps going. The server will start a timer every turn of the disconnected client and if the client doesn't send a package in time, that client's turn is skipped. When the last client in a match disconnects, the game will ask them if they want to save the game and resume a later time; the client sends their answer and the server acts accordingly, closing the connection.