Peer-Review 2: Network protocol

Davide Preatoni, Federico Sarrocco, Alessandro Vacca
GC30 Group

May 4, 2022

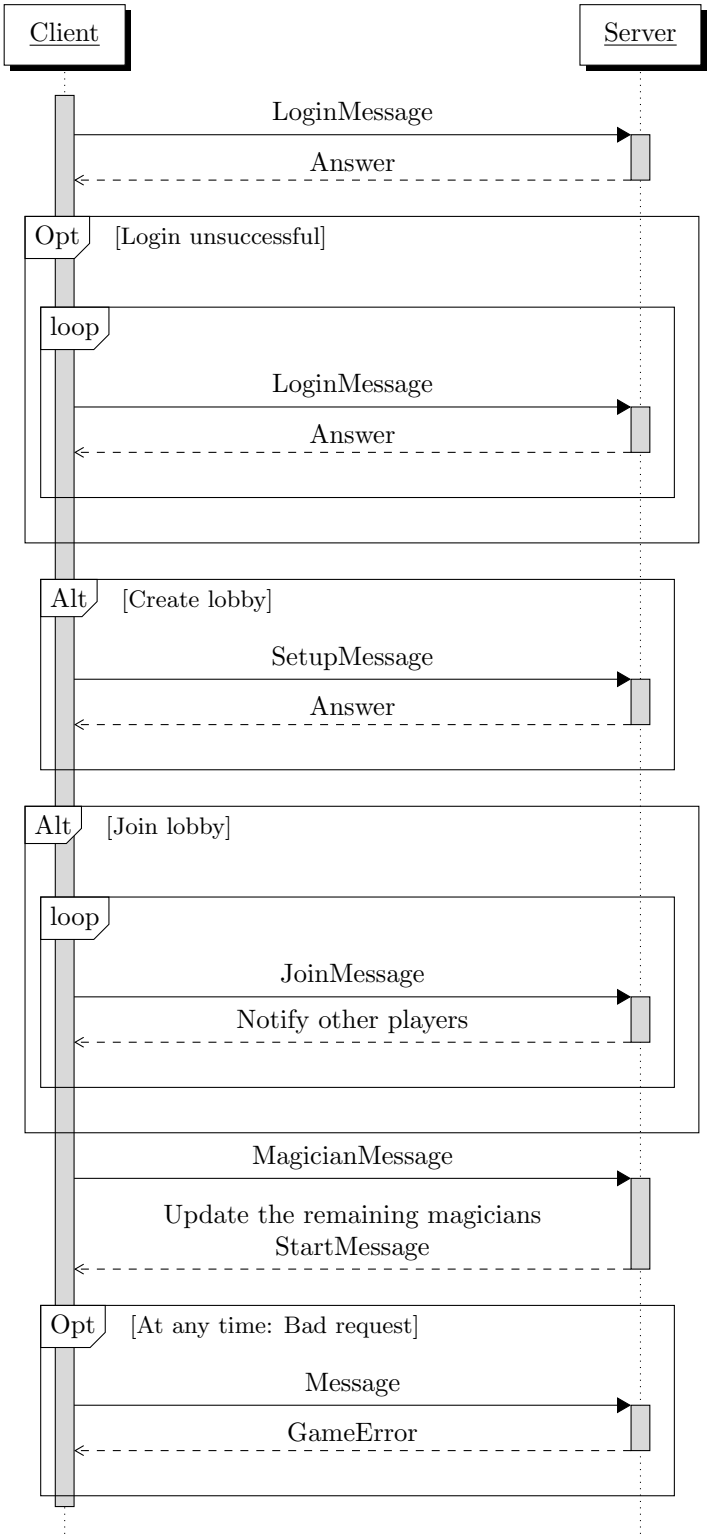
As suggested by Prof. Cugola, we wrote this document with the intent of further clarifying our UML diagram by explaining the network protocol with a brief descriptions and UML Sequence Diagrams.

1 Starting the game

After the server has been started with its own port number, a client can start a connection by providing the server's port number and IP address. After having accepted the request, the server waits for a Message of the type LoginMessage, that contains the nickname that will be assigned to the player, otherwise it answers with a GameError. After receiving the nickname, the server deserializes the data and parses the corresponding nickname, after testing for its uniqueness and presence. If the specified nickname is already present the server will wait for until a correct LoginMessage is received. If the client specifies a valid nickname, the server waits for a Message either of the SetupMessage type or the JoinMessage type.

- **Lobby creation:** the supplied SetupMessage contains the lobby creation data, as in the number of players to be included in the game and if the game is set to the expert mode. The server parses the data, creates the lobby, adds the player and waits for the other clients.
- **Lobby join:** the received JoinMessage contains the ID of the specified lobby that the player wants to join. The server parses the data and then adds the player to the correct game lobby.

When a lobby reaches its specified number of players (either 2 or 3) the server automatically starts the game setup. The players will now select their Magician cards, one at a time. The lobby creator first selects its Magician, and sends the selection to the server, which then updates the remaining Magicians on the other clients. When the last player selects its Magician, the server updates all the clients with a StartGame message, that contains the list of players and their respective Magicians.

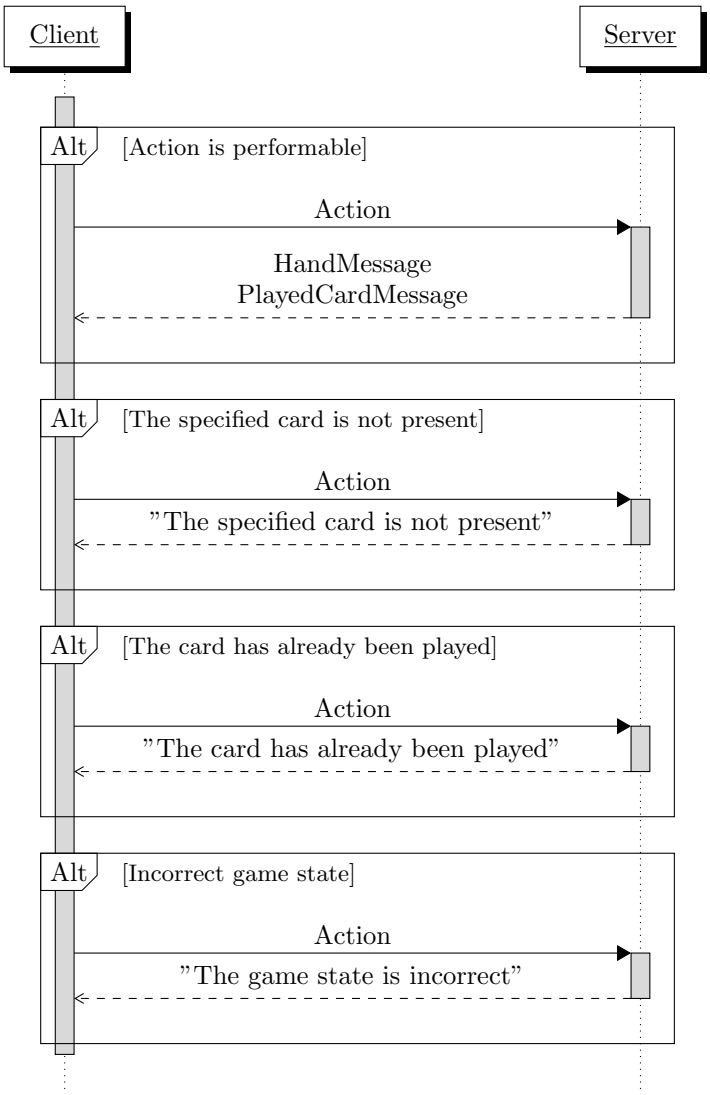

Client
Server
LoginMessage
Answer
Opt
[Login unsuccessful]
loop
LoginMessage
Answer
Alt
[Create lobby]
SetupMessage
Answer
Alt
[Join lobby]
loop
JoinMessage
Notify other players
MagicianMessage
Update the remaining magicians
StartMessage
Opt
[At any time: Bad request]
Message
GameError

2 Actions

In every turn, the available moves to the client will be exposed by the current GameState (for every state, there are certain actions available).

2.1 Play an assistant card

The player selects the card that he wants to play via CLI or GUI and the selection is then parsed to an Action message with the corresponding type ("PLAY_CARD") and an integer number equivalent to the selection. The message then gets serialized an then sent to the server, that parses the action to the corresponding move on the Server's model, that gets changed, and the changed is then propagated to the VirtualClients on the Server, that then forward the corresponding updates to the clients via the network. The updates are HandMessage, that sends that updated hand of the player, and PlayedCardMessage, that tells all the clients which card has been played. After all the players have played their card, the game moves to the next phase.



2.2 Select a cloud

As for the card selection, the player chooses the cloud he wants to get via CLI or GUI; the choice is then parsed into an Action message with the corresponding type ("CHOOSE_CLOUD") and an integer number equivalent to the selection. The message is serialized and then sent to the server, which parses the Action action to the corresponding move on the server model, which gets modified, and the modifications are propagated to the VirtualClients on the server, which then forward the corresponding update to the clients over the network. After the players has chosen his cloud, the game moves to the next player's action phase.