Peer-Review 2: NETWORK PROTOCOL DESIGN

Valeria Amato, Francesco Barisani, Nicolò Caruso

Gruppo AM38

Network protocol and sequence diagram evaluation of the group AM11

Positive aspects

- Convenient choice of using a Timer between the Client - Server interaction for the HeartBeat / Ping-Pong.
  Using a timer between the normal interactions between the Client and Server avoids using separate threads for the Heartbeat alone. Instead of a separate thread, the ping pong messages are activated after the timeout and received by the Server.

- Simple and atomic messages with only the required fields and values that also allow a less granularity on the control of the possible errors caused by the message.
  This choice allows more control on the message exchange and the possible issues between these network exchanges.

- As for the PlayerActions sequence diagram, it is effective to repeat the “opt” block for the usage of the Character cards multiple times during the Action phase, in order to model the fact that a Character card could be used either before or after moving the Students.

- Good coverage of error message types that allows the identification of the error uniquely and react to them with the right procedure instead of more generic errors.

Negatives aspects

- In general, the Error messages could contain some additional fields indicating the allowed ranges of values for the input. For example, the “Character not usable” message should be followed by some kind of list with all the usable Characters.
  The same could be said for the Card messages. There should be a list of the available Cards for every player.

Login Phase :

- It is unclear how the parameters are sent in the Login phase.
  The sequence diagram shows them as separated sequential messages, whereas the doc and the json messages samples show them packed into a unique message.
  Anyhow, we would suggest sending them as separated messages, since it allows to better handle the setup game logics

- The Loop “Lobby filling” where the Server keeps sending “Waiting for players” should be modified because the Server shouldn’t send messages to the clients without any previous prompt from the Clients

- From what we see in the Sequence Diagram, it is unclear what is the interaction from the Client that finally triggers the “Creating game” and “show board created” messages sent by the Server. They seem to be unprompted messages sent by the Server, while probably they are solicited by the last login.

Player Actions Phase :

- In the sequence diagram of PlayerAction messages, we see some Update messages, but there is no corresponding format of those messages in the txt file with all detailed messages. Therefore, we would suggest specifying the fields and the content of those Update messages.
  We suggest different kinds of Update messages, ranging from simple updates/notifications to more informative updates like whole classes or maps and lists based on your model structure.

- In the MoveStudent message there should be a field indicating the Color of the moved student

- Considering the rules of the game, the choice of the Cloud should not be modeled at the beginning of the action phase, but should occur at the end of each turn.

Comparison between the two protocols

- Use of timer between interactions. If it elapses → show error
  We did not model this type of error message or action for the interaction with the client but its implementation seems to be needed in order to avoid situations where the client just stalls or waits in its active phase. This particular solution, soliciting the player, takes also into account a possible away from keyboard situation and could also support an automatic disconnection/change of turn due to inactivity.

- Interaction for Mage selection
  Here, the login phase also includes the necessary message exchanges to choose the mage deck. While we have modeled the choice as automatically done by the controller.

- Multigame
  This network implementation takes into account the multigame advanced functionality since there are no errors due to full lobbies. While our solution has a unique lobby.

- This approach seems to use conversions of the messages as structured classes, while our approach uses more common fields to identify the successive properties and values of the json message.

- The sequence diagram models well the computations/calculations that occur within the Server before and after the messages exchanged with the Client, such as “Dominance calculation”, “Merge Calculation”, “Check win condition” . It helps to understand better the logic flow of the Action phase.
  We did not use any of these loops in the sequences, but we could include this approach with some interaction loops that occur within the Server.