Peer-Review 2: Networking

Andrea Torti, Diego Viganò, Cristiano Valtolina, Fabio Vokkri
Group GC33

May 5, 2023

Evaluation of the Networking part of GC23's project.

1 Positive aspects

Firstly, we find the use of textual commands within the TUI by prefixing a slash to the input string to be very effective.

In addition, we believe that the ability to change one's nickname, create custom lobbies, and access specific lobbies as indicated by the client, although not explicitly requested within the project specifications, are useful ux-oriented features and are designed in a simple and effective manner.

2 Negative aspects

Since the game is not timed, we believe that asking a client to start the game is redundant: the game can start immediately as soon as the last client enters the lobby.

The move information can be condensed into a single message: this would also simplify the controller and reduce the number of methods it has. This implies that the move validity checks are also executed client-side.

In the UML diagram of the server part, we do not notice the presence of run() methods: it is not clear which classes are runnable.

It is not clear from the network UML where the information in the messages is stored, even though we are told in the summary document that it is inside them, which are serializable objects.

Since there are updates but no notifies, it is not clear if the listener pattern has been used and therefore which classes are observable.

3 Comparison between architectures

The other group seems to have clearer ideas and has produced better documentation regarding the registration, entry into a lobby, and setting of usernames by players, while we have focused more on the main gameplay flow. We find the idea of prefixing a slash to commands to be sent to the server, already seen in other popular video games, to be very good, so much so that we want to somehow imitate it. It could be useful for asynchronously sending chat messages via TUI during other players' turns. We refrain from commenting on the RMI part, as we have yet to implement it in our project.