Peer-Review 2: Sequence diagram

Corona Mattia, Grisanti Davide, Mori Tommaso, Spelta Francesco Giuseppe

Gruppo AM38

Valutazione del sequence diagram del gruppo AM47

Lati positivi

- **THREAD:** the diagram shows the differentiation between clients, server and LobbyThread which gives an immediate idea on how the processes work together (ex: using a timer for the game start). All of this is very well documented in the first two pages of the attached document.
- **GSON:** the format used to share information between the server and clients is shown in the diagram, more details about this are also shown in the first page of the attached document.
- **LOBBY NAMES:** using a custom name for the lobby, chosen by the host, it's a nice idea.

Lati negativi

- **UNCLEAR ARROWS USAGES:** it's unclear what some of the arrows mean, the ones that start and end from the same component (either LobbyThread or ServerMain), some seem to be private methods others look like calls to some methods of the Model. Maybe dividing Controller and Model could help read those arrows/calls in a much easier way.
- **METHODS/PARAMETERS:** unspecified signature for the methods referenced in the diagram, both methods names and their parameters aren't clear (types and names missing).
- **ENDGAME TRIGGER:** the endgame logic is unclear/missing : in the attached document not all the game rules are included (missing endgame trigger: when the decks are empty), furthermore in the diagram some others are seen as given: once a player scores their 20^{th} point (or the decks run out of cards) each player, that has not taken a turn in that round yet, still needs to play a turn before the last round and shouldn't receive the "FinalTurn" notification.

Confronto tra i diagrammi

SIMILARITIES

- Login management.
- Considering nicknames as globally unique.
- We both implemented the same advanced functionalities and managed them in very similar ways.

DIFFERENCES

- Identifying a lobby by a custom name was a fun idea, we instead opted for the usual (and boring) game ID, which will be assigned automatically, instead of being chosen by the host.
- We partitioned our sequence diagram in View/Client-Controller-Model, explaining in detail the methods called by each part on any other one (usually the Client calls a method on the Controller by interacting with their chosen View, and then the Controller calls the needed methods on the Model by following and translating the Client requests).