

Peer-Review 2: Protocollo di Comunicazione

**Endi Kucuku, Giovan Battista Landolina,
Alen Kaja - Gruppo 49**

Valutazione della documentazione del protocollo di comunicazione del gruppo 48

Lati positivi

- Riteniamo che il design della comunicazione client-server proposto dal gruppo sia complessivamente positivo. Sono stati infatti mostrati numerosi scenari come il *Login* dell'utente, il *Game Start*, la *Planning Phase*, l'*Action Phase*, l'*End of the game* e la *Update view*. Essi consentono di riassumere le principali fasi del gioco.
- Ognuno dei messaggi scambiati è stato documentato e commentato in riferimento ai parametri che necessariamente verranno passati. Riteniamo positivo il fatto di aver catalogato i tipi di messaggio per argomento e per risposta possibile.
- Gli scenari sono stati descritti con la corretta sintassi dei sequence diagram e risultano comprensibili e chiari. Inoltre sono ben distinti, dato che assolvono a diverse situazioni.

Lati negativi

- Alcuni messaggi potrebbero richiedere altri parametri che saranno necessari al momento dell'implementazione.
- Il client dovrebbe comandare la partita e il server essere il più possibile reattivo, invece di direzionare il gioco fornendo i comandi da effettuare: ad esempio una

volta iniziata la partita il client decide quale carta giocare e il server permette o meno la scelta.

- In alcuni casi può essere ridondante l'*acknowledgment* da parte del client: nel caso dell'update della view o della partita terminata il client può semplicemente ricevere il messaggio e constatare il cambiamento di stato o evoluzione del gioco.

Confronto tra le architetture

- In confronto al nostro protocollo, il gruppo 48 presenta un diagramma di messaggi più ad alto livello e non differenzia ancora i vari strati di Client e Server. Secondo noi i messaggi transiteranno sul *socket* per poi arrivare a delle classi che fanno da "controller".
- Il diagramma del gruppo 48 sembra essere più "snello" del nostro visto che presenta varie situazioni isolate. Potremmo adattare il nostro secondo questa linea in modo da presentare meglio i vari scenari.
- Anche noi similmente gestiamo la risposta del server tramite una logica di *acknowledgment*.