

# Peer-Review 2

Carlo Sgaravatti, Jacopo Maria Polverini, Luca Romano

Gruppo GC34

Valutazione del diagramma UML delle classi della parte di rete e del protocollo di comunicazione del gruppo GC44.

## Lati positivi:

- L'utilizzo della VirtualView, in quanto permette di gestire in maniera pulita i messaggi che viaggiano dal client al server e viceversa.
- L'utilizzo dei listeners, perché permettono di gestire al meglio gli eventi lanciati in seguito ad una modifica del model e di notificare facilmente ciò al client.

## Lati negativi:

- Mandare l'intero field al client è piuttosto pesante e ripeterlo ad ogni modifica potrebbe risultare inefficace. Potrebbe essere più opportuno salvare alcune informazioni sui client e aggiornare dopo le varie mosse solo le modifiche effettuate (in questo modo ogni client ha anche chiaro quali sono state le modifiche e a seguito di quale mossa sono state fatte le modifiche). In aggiunta, l'invio dell'intero field espone il model al client e quindi alla view (nonostante il field venga clonato, il client viene comunque a sapere come è strutturato il model, cosa che nel pattern MVC si vorrebbe evitare in quanto la view è indipendente dalla struttura del model).
- Inviare il field ogni volta può comunque andare bene per la CLI ma è scomodo per la GUI (ogni volta andrebbe rifatta l'intera scena, il che potrebbe dare problemi di prestazioni).
- Non sono stati previsti messaggi di acknowledgement in risposta a nessun tipo di messaggio che il client manda, in questo modo il client non può sapere se la mossa che ha richiesto di fare è andata a buon fine oppure no. Inoltre, non sono stati previsti messaggi di verifica della connessione (ping), il che potrebbe non bastare in quanto il protocollo TCP (poiché è un protocollo affidabile) continua ad aspettare pacchetti dal client pensando che la rete sia

congestionata anche se questo si è scollegato (quindi il server potrebbe non accorgersi che il client si è disconnesso).

- Mancano i messaggi per la scelta delle torri e dei maghi all'inizio del gioco; inoltre, non sembra essere previsto un messaggio per la creazione della partita (il primo giocatore che fa il login deve poter scegliere il numero di giocatori e il tipo di regole).

## Confronto tra le architetture

Il ClientHandler e la VirtualView sono molto simili a come noi abbiamo pensato di gestire la comunicazione con il client e l'aggiornamento del model in seguito ad una mossa proveniente dal client. Inoltre, anche la gestione degli eventi con i listener è molto simile a come l'abbiamo implementata noi.