

Peer-Review 2: Rete e Protocollo

Emanuele Paci, Lorenzo Paleari, Thomas Puppinato
Gruppo 12

9 maggio 2022

Valutazione dell'architettura di rete e del protocollo del gruppo 22.

1 Lati positivi

Osservando l'UML e i sequence diagrams forniti, abbiamo individuato alcuni aspetti positivi, quali una buona e corretta implementazione del pattern MVC, la realizzazione effettiva dello strategy pattern, una buona suddivisione degli eventi e infine una buona gestione del multipartita e della comunicazione tra client e server.

Abbiamo notato la corretta implementazione dello Strategy pattern (che prima era solo intuibile dai commenti delle carte personaggio) volto a modificare le regole relative al calcolo dell'influenza, riuscendo così a raggruppare 4 effetti di personaggi diversi con un unico Strategy.

Sono presenti eventi per ogni tipo di azione di gioco richiedibile dal client che vengono sfruttati in maniera intelligente. Come si può notare dal Sequence Diagram (1) e come richiesto, è il client a richiedere di eseguire un'azione, con conseguente risposta da parte del server.

Dal Sequence Diagram (2), invece, abbiamo notato l'implementazione della funzionalità avanzata del multipartita, apprezzabile l'idea di chiedere ad ogni giocatore il numero di persone con cui vuole giocare per poi mettere assieme coloro che hanno scelto lo stesso numero.

2 Lati negativi

Nel Sequence Diagram (2) non è specificato quando e se venga chiesto all'utente se vuole iniziare la partita in modalità standard o expert; essendo una scelta definita dalle regole di gioco, è necessario rappresentarla esplicitamente, così da poter associare, oltre in base al numero di giocatori, anche persone che vogliono giocare con la stessa modalità.

A livello di UML manca la parte di client, la cui struttura è intuibile solamente guardando i Sequence Diagram.

3 Confronto tra le architetture

La struttura dell’UML, nel complesso, è simile alla nostra. A differenza vostra, abbiamo deciso di implementare tutti e 12 i personaggi, abbiamo poi una singola VirtualView (RemoteView) con dei riferimenti a tutti i clientHandler.

Inoltre, abbiamo deciso di usare la serializzazione presente in java al posto di file JSON: tale scelta ci ha facilitato il passaggio di classi “Messaggio”, che implementano tutti i metodi di una stessa interfaccia, favorendo overriding e overloading.

Abbiamo scelto come voi di implementare la funzionalità aggiuntiva del multipartita, dando la possibilità al client di scegliere in quale lobby entrare mettendogli a disposizione il nome dei giocatori che si sono già connessi, creando una lobby, oltre ad alcune informazioni utili sulla modalità di gioco (numero di giocatori e expertMode).