

Peer-Review 2: Network

Stefano Riva, Andrea Tonoli, Francesca Dora Pieruz, Edoardo Sartore

Gruppo C37

Valutazione del diagramma class diagram del network e sequence diagram del gruppo GC27.

Lati positivi

L'interfaccia VirtualServer è un'ottima soluzione per gestire il caso in cui due client della stessa lobby vogliano connettersi utilizzando due tipi di connessioni diverse. Un altro lato positivo è la numerazione delle lobby in quanto nel caso di disconnessione del Client, quest'ultimo ha la possibilità di riconnettersi a patto che conosca il numero della lobby in cui stava giocando.

Lati negativi

Sono presenti delle discrepanze tra i metodi e le classi del class diagram e dei sequence diagram (ad esempio nel sequence diagram la classe ServerApp invoca il metodo runServer() che è invece presente nelle classi RmiServer e SocketServer). Inoltre, pensiamo che un'altra problematica è data dal fatto che è la TUI e non il Client a "dialogare" con il VirtualServer, questo risulta essere un problema perchè lo scopo della View in generale è quello di visualizzare la situazione del Model e non di comunicare con il Server.

Confronto tra le architetture

Abbiamo notato delle differenze a livello implementativo per quanto riguarda la creazione della TUI, nel nostro caso viene creata immediatamente prima del login del Client, nel vostro caso invece la TUI viene creata dopo il login del Client. Anche per la connessione Socket abbiamo notato delle differenze, in particolare noi gestiamo la connessione con degli appositi messaggi della classe Message, mentre voi avete scelto di implementarlo in modo più simile a RMI come se fosse un'invocazione di metodi. Un consiglio che possiamo dare è di utilizzare una classe apposita per i messaggi nella comunicazione via Socket al posto di usare delle String.