

Peer-Review 2: UML e Network

Pietro Roggero, Lorenzo Tallarico, Matteo Trombetta, Tommaso Razza | Gruppo GC06

Valutazione del diagramma UML delle classi del gruppo <CG53>.

Lati positivi

Buona l'idea per l'implementazione della chat, si consiglia di aggiungere un campo destinatario in quanto la specifica di progetto richiede la possibilità di invio di messaggi privati al singolo client. Buona l'idea per gestire eventuali disconnessioni da parte del client e monitorare i client attivi.

Lati negativi

L'UML risulta incompleto e, ai fini di questa peer review, non risulta comprensibile come avete deciso di implementare la rete. Non sono presenti attributi in nessuna classe, consigliamo di aggiungerli per le documentazioni successive. Non è chiaro come il Game Controller si interfacci con server e client, soprattutto non si capisce la scelta progettuale di implementare gli stessi metodi sia sul server che sul client. La classe ClientCallBack è vuota, è definita come interfaccia ma non è chiaro quale sia il suo scopo. Le parti di codice scritte per RMI sono inconsistenti in quanto i parametri vengono passati in maniera errata.

La descrizione di PerformGameAction non ci dà modo di valutare il suo funzionamento.

Confronto tra le architetture

Sia la nostra architettura che quella del gruppo revisionato implementano la tecnologia RMI per poter implementare la comunicazione tra client e server. In caso di una futura implementazione anche della tecnologia Socket si consiglia di astrarre le classi già esistenti in modo da poter separare la logica di gioco dalla tecnologia di connessione.

Ipotizziamo che la classe ClientCallBack sia stata creata per notificare eventuali modifiche del model ai client, se così non fosse, consigliamo di implementare l'utilizzo di listeners per rispettare il pattern Model-View-Controller.