

Peer-Review 1: UML

<Rosano Francesco>, <Rutigliano Davide>, <Sangalli Matteo>, <Touati Amin>


Gruppo 14

Valutazione del diagramma UML delle classi del gruppo 04.

Lati positivi

- L'attenzione e la precisione permettono di avere sotto controllo tutti gli eventi possibili e poter gestire correttamente ogni possibile problema.
- Corretta la scelta di dare trasparenza alla rete tramite un'interfaccia poi implementata dai due protocolli di rete.
- Ottima idea l'utilizzo di un segnale di ping per tenere controllato lo stato dell'utente.
- Buona la gestione delle interfacce per rendere anche estendibile la serializzazione di nuove classi per nuovi messaggi con l'override di GameControllerInterface execute(GameListener lis, MainControllerInterface mainController) e void execute(GameControllerInterface gameController), con buona anche l'idea di passare 'lis' per avere un oggetto remoto Client per il Server.

Lati negativi

- Come sottolineato anche nella precedente revisione, l'eccessivo dettaglio riduce la leggibilità del diagramma UML.
- I messaggi inviati dal client al Controller presentano alcune criticità:
 - Secondo le regole si viene aggiunti ad una partita a caso, la si crea solo se non ci sono altre partite in cui entrare. Quindi dovrebbe esserci solo un messaggio di join a seguito del quale il Main Controller crei una nuova partita o aggiunga il giocatore ad una partita già esistente
 - Reconnect potrebbe essere incorporato in join con un flag, magari di tipo booleano, che indica se la join sia una riconnessione o meno.
 - IsThisMyTurn ci sembra superflua: il server può mandare un messaggio di inizio turno che "attivi" il client nella gestione del suo turno.
 - playerIsReadyToStart ci sembra superflua: come avviene in svariati giochi online, il server manda un messaggio di inizio turno e se non riceve alcuna risposta a distanza di un determinato intervallo passa a gestire il turno del giocatore successivo.
 - Il significato delle iconcine foglio nel sequence diagram, ovvero  risulta intuibile, ma il loro significato preciso sfugge sicuramente. Andrebbero spiegate anche solo brevemente per non perdere dettagli.
 - Sempre nel sequence diagram, perchè in alcuni dispatch (in alcune interazioni) si dà un gameModelImmutable come parametro?

- Abbiamo compreso l'idea di permettere al controller l'utilizzo dei due protocolli di rete senza sapere quale si stia utilizzando, però pensiamo che questa divisione non vada applicata nei listener ma in modo diverso, come spieghiamo nel paragrafo sottostante.

Confronto tra le architetture

- Il gruppo 04 ha un maggior grado di dettaglio che garantisce loro una maggior precisione riguardo al riconoscimento ed alla gestione dei vari messaggi, rispetto a noi che dovremo essere più dettagliati nell'identificazione di pacchetti ed eventi.
- Entrambi i gruppi hanno pensato di garantire trasparenza della rete dal punto di vista del controller. Noi abbiamo deciso che gli scambi fra Server e Client avvengano tramite singoli pacchetti d'informazioni che poi, tramite uno strategy pattern, vengono spediti secondo il protocollo di comunicazione scelto dal player. In questo modo il controller potrà utilizzare un unico metodo "send" che garantisce completa trasparenza della rete. La scelta del gruppo 04 ci sembra garantisca solo in parte questa trasparenza.
- Entrambi i gruppi hanno pensato di utilizzare un segnale di ping per verificare che i player restino connessi durante la partita [Gestione delle collisioni lato client]. Tuttavia, c'è differenza riguardo il tempo fra un segnale e l'altro: noi prevediamo che il server mandi il segnale ogni 3s, in caso per 5s non riceva niente manda un ulteriore segnale sottolineando l'assenza di risposta, dopo 10s di assenza di risposta il client è considerato offline.
- Il gruppo 04 usa le classi GameListenersHandlerSocket e GameListenersHandlerClient per i rispettivi socket client e server, in modo da distinguere i metodi Listeners per ciascun socket. Noi li avremmo accorpati nella classe ClientHandler e ClientSocket, accorpando i metodi necessari e "risparmiando" classi.