Peer-Review 2: NETWORK

Leonardo Ratti, Francesco Rivitti, Denis Sanduleanu, Alessandro Salvatore

Gruppo 30.

Valutazione del diagramma UML delle classi del gruppo 20.

Lati positivi

Indicare in questa sezione quali sono secondo voi i lati positivi dell'UML dell'altro gruppo. Se avete qualche difficoltà, provate a simulare il gioco a mano, immaginandovi quali sono le invocazioni di metodo che avvengono in certe situazioni che vi sembrano importanti (ad esempio, la fusione delle isole oppure il calcolo dell'influenza).

- Buona compartimentalizzazione dei messaggi: molto comodo usare un'interfaccia alternativa per i messaggi di sola notifica ai player, come anche differenziare i messaggi da quelli all'interno della partita e quelli al di fuori dalla partita.

Lati negativi

Come nella sezione precedente, indicare quali sono secondo voi i lati negativi.

- Si parla di RequestHandler come interfaccia implementata da ServerManager, ma non ce n'è traccia nell'UML.

- LocalView deve essere serializzata e inviata a tutti i client al fine di aggiornare la view di ognuno, ma in LocalView non c'è traccia di informazione riguardante il model.

- Non condividiamo la scelta di inviare 3 messaggi diversi (con relativa risposta dal server) poiché la quantità di informazione che si invia è poca, si riuscirebbe a ridurre il numero di comunicazioni, arrivando anche a un solo messaggio (e anche ovviamente la conferma del server).

- Non si comprende appieno il ruolo della classe StaticStrings.

- Non ha senso l'invio di messaggi di ping perchè attraverso l'implementazione di Socket e RMI nel caso in cui cadesse la connessione da parte del Client, lato server ci sarebbe una exception (inoltre l'arrivo stesso del messaggio rappresenta un ping).

Confronto tra le architetture

Individuate i punti di forza dell'architettura dell'altro gruppo rispetto alla vostra, e quali sono le modifiche che potete fare alla vostra architettura per migliorarla.

- L'architettura inviataci si preoccupa di inviare l'azione del player giocante divisa in più messaggi: in particolare si inviano due (o tre) messaggi diversi per l'intero turno

giocato, cioè scelta delle tessere, ordine di queste e scelta della colonna; noi invece operiamo solo un invio con l'intero pacchetto delle informazioni: questo è controllato sia lato client, prima di essere trasmesso, sia lato server per assicurarsi che il gioco proceda secondo le regole.

- Voi implementate una classe WaitingRoom che rappresenta una lobby specifica, con tanto di ID della lobby; noi non abbiamo una classe simile, decisione guidata dal fatto che non stiamo implementando le partite multiple.