Peer-Review 2: Network UML

Maddalena Panceri, Alessandro Ruzza, Gabriele Santandrea, Flavio Villa

Gruppo 33

Valutazione del diagramma UML delle classi del gruppo 23.

Lati positivi

Nonostante noi abbiamo scelto di rendere i protocolli asincroni, riconosciamo la solidità della comunicazione sincrona, specie per la rilevazione di problemi di rete.

Altro lato positivo è l'alta capillarità dei messaggi e la compressione delle informazioni dove possibile, (e.g. inviare solo il tipo delle top card dei deck in UpdateSharedPools).

Ottima la gestione della classe Client che interfaccia Network e View.
L'uso di un LocalGame per avere una versione "leggera" del model in locale al client è anch'esso una scelta vantaggiosa.

Lati negativi

MESSAGGI

I messaggi di UpdateHand, UpdateSharedResources, UpdateSharedPools non sembrano essere differenziali. Inoltre, inviando gli oggetti carta, potrebbero rendere pesante la comunicazione su rete (soprattutto se sono in corso molte partite in contemporanea).

Con l'uso dei messaggi, il protocollo RMI è stato "ridotto" in astrazione, avvicinandosi a TCP e perdendo le potenzialità di parallelismo e leggibilità di rmi.

CLIENT-SERVER

Il metodo setModel() da chiamare dal server verso il client ha come parametro LocalGame a cui il Server non ha accesso.

Tutti i metodi di VirtualView possono essere chiamati solamente dal Client verso il Server, quindi è possibile che vi sia stato un errore nella stesura del documento di descrizione.

I metodi handle, getUname e setUname sono irraggiungibili al di fuori delle classi SocketClient e RmiClient. Quindi sono metodi privati. RmiClient non ha una getter del model a differenza del SocketClient.

Non esiste una funzione di ping per la versione Socket.

Il SocketServer e RmiServer sono unici per tutte le partite. Questo può causare rallentamenti

Il controller non dovrebbe inviare informazioni al client. RequestUpdateObservable dovrebbe essere nel model. Inoltre era consigliato avere più observable per diverse parti del model.

Confronto tra le architetture

Ci manca una funzione di closeConnection dal gruppo implementata nel metodo socketDie()

Noi consideriamo LocalGame e Client come parte della View, ma potremmo spostarla nella parte network, delegando alla View solo il compito di visualizzare efficacemente le informazioni a schermo.