

Peer-Review 2: Protocollo di rete

Pressiani Giuditta, Rossi Gabriele, Spinelli Davide, Travaini Alessandro

Gruppo GC-19

Valutazione del protocollo di rete del gruppo GC-09.

Lati positivi

Vista l'assenza di un controller anche a lato client, l'architettura risulta più veloce e leggera. La Lobby, che gestisce tutte le interazioni tra la parte Model-Controller e View tramite lo strato di rete, astrae maggiormente la rete dalla logica applicativa e permette una maggiore estendibilità.

La classe GameUpdate che mostra la partita allo stato corrente è una buona idea per garantire che il client riceva sempre la partita aggiornata ad ogni turno.

Lati negativi

La lobby, nonostante sia la parte centrale di comunicazione tra Client e Server, non è chiaro dove risieda leggendo il diagramma UML e il commento allegato. Ipotizziamo per intuito che risieda lato server.

Riguardo la classe GameUpdate, l'invio ad ogni turno degli obiettivi personali e comuni risulta ridondante, visto che non cambiano durante tutta la partita; si potrebbe semplificare questo scambio, inviando gli obiettivi solo all'inizio della partita e facendoli salvare lato client, in modo tale da permettere all'utente di visualizzarli quando necessario, senza l'utilizzo della rete.

Confronto tra le architetture

Mettendo a confronto le due architetture, non si può non notare che l'architettura del gruppo 09 è più snella e diretta; il nostro gruppo utilizza la comunicazione RMI come base di tutto il protocollo di rete, lasciando come comunicazione "fantoccio" quella socket, che ci si appoggia totalmente.

Pensiamo sia una buona idea servirsi di messaggi, anche da noi adottati, per mandare le azioni tra client e server.

Al contrario dell'uso della Lobby, abbiamo invece creato una "virtualview" in modo tale da mascherare la rete al controller di gioco, ma comunque permettendogli di comunicare con l'esterno aggiornando una view fasulla.

Concludendo, vista la semplicità dello strato di comunicazione, proveremo a snellire le interazioni tra client e server e di conseguenza quelle tra Model-Controller e View.