

Peer-Review 2: Network

Karanbir Singh, Kevin Thomaj, Gabriele Tonello, Lara Tongiorgi

Gruppo GC26

Valutazione dell'architettura di rete del gruppo GC16

Lati positivi

Model

Appreziamo molto la maggior semplicità del Model rispetto la scorsa volta, che risulta più ordinato e leggibile.

Architettura di rete

La gestione del GameAction sembra implementata bene. Un'altra nota positiva è la presenza di un PlayerController per ogni giocatore perché rende più modulare e indipendente le diverse gestioni delle richieste del client. Ottima la scelta di utilizzare il design pattern Observer - Observable per l'implementazione e utilizzo delle notify.

Lati negativi

Connessione e creazione match

Nel join di una partita già avviata, è ridondante avere due metodi separati come la connect e la join, si potrebbero collassare in un unico metodo in modo da rendere la procedura più veloce.

Inoltre fornire al Client tutte le partite disponibili non dà molte informazioni per poter scegliere a quale Game unirsi, mentre avrebbe senso utilizzare un codice di ingresso. La gestione di queste richieste potrebbe creare problemi nel momento in cui diversi Client cercano di collegarsi contemporaneamente alla stessa partita, quindi se non l'avete già implementato, consigliamo di aggiungere dei controlli in modo tale da garantire il rispetto del numero massimo di giocatori.

Da quanto abbiamo capito, l'avvio della partita è chiamato dall'admin con il metodo start, nel caso voglia giocare con meno di 4 persone. Però, siccome è un metodo chiamato da un Client, tra la chiamata del metodo e l'effettiva esecuzione potrebbero entrare altri giocatori. Potreste gestire questo problema permettendo all'admin di impostare al momento della creazione della partita il numero di giocatori con cui vorrà giocare e far avviare la partita non appena viene raggiunto.

Gestione tecnologie di rete

Per poter implementare anche la possibilità di connessione TCP, mancano delle interfacce generali per le classi relative alla parte di network, che poi andranno implementate in modo diverso sulla base della tecnologia usata.

Per quando implementerete la parte di socket, tenete a mente che le due tecnologie di rete dovranno essere o entrambe sincrone o entrambe asincrone. Per rendere asincrona la tecnologia RMI potreste valutare l'utilizzo di una coda di richieste, in modo da evitare potenziali blocchi di concorrenza.

Confronto tra le architetture

Nella nostra architettura abbiamo fatto ampio uso delle interfacce, in particolare per l'implementazione delle classi di rete, in modo tale da nascondere la tecnologia di rete scelta dall'utente e generalizzare le chiamate dei metodi.

Potremmo prendere esempio dalla vostra implementazione delle notify in quanto ci sembra ben strutturata e semplice.