

Peer-Review 2: Architecture

Francesco Palumbo, Matteo Pancini, Luigi Pascali

Gruppo GC31

Valutazione della parte di architecture del gruppo GC41.

Lati positivi

- La creazione di un unico oggetto `PlayerAction` risulta comoda, permettendo infatti di gestire in maniera univoca tutte le azioni che possono essere scelte dal giocatore
- L'interfaccia di rete è molto semplice grazie alla gestione dei vari metodi attraverso la sola classe `Connection`

Lati negativi

- Se da un lato è apprezzabile l'unicità della classe `PlayerAction`, dall'altro risulta a nostro parere un po' eccessiva, in quanto molte azioni non sfrutteranno gran parte degli attributi che sono stati messi. Potrebbe essere infatti altrettanto chiaro e semplice creare varie classi per ogni azione, così da gestire i singoli attributi solo laddove necessari.
- Non ci è totalmente chiaro l'utilizzo dell'oggetto serializable `ModelClone`
- Utilizzare pochi oggetti per la gestione delle connessioni potrebbe rischiare di rendere più pesanti (e complicati) gli oggetti stessi creati in quanto ricchi di metodi

Confronto tra le architetture

Le nostre architetture risultano, ovviamente, simili seppur gestite con un differente approccio.

Nel nostro caso si è sfruttato un numero maggiore di classi, in modo tale da suddividere maggiormente le varie operazioni, mentre nel caso del gruppo GC41 si è semplificata la struttura a oggetti, inserendo più elementi all'interno di un numero inferiore di classi.

Anche dal punto di vista dei messaggi che attraversano le socket il gruppo GC41 ha prediletto un'unica classe che verrà poi modificata in base all'azione scelta dall'utente. Nel nostro caso abbiamo invece preferito generare dei messaggi ad-hoc per ogni possibile azione, in modo tale da snellire non solo le singole classi delle azioni, ma anche la loro gestione lato client-server.