

Peer Review 2: Sequence Diagrams

Revisione del Gruppo AM18 da parte del gruppo AM09

Aspetti positivi

Tutti i sequence diagram sono ben strutturati, rappresentano logiche di funzionamento corrette ed ogni input del client è validato dal server.

Quindi le nostre osservazioni si riferiranno a scelte progettuali e a stile e non a comportamenti scorretti dell'applicazione.

Aspetti negativi

Il fatto che la risposta della connessione sia un integer rende i sequence diagram, e probabilmente anche il codice, difficile da interpretare senza la presenza di commenti o annotazioni.

Riteniamo che sarebbe una pratica migliore creare una enum `ConnectionResponse`, con un nome più "parlante", in modo da rendere più chiari la codifica dei messaggi e lo stato dell'applicazione a seguito della risposta.

Relativamente all'input del client riteniamo che sarebbe più efficiente costruire un oggetto `Move` lato client che contenga tutte le informazioni da passare al controller del server per modificare lo stato della partita. Questo oggetto potrebbe essere costruito progressivamente lato client validando un input per volta ed essere validato in un unico passaggio dal server una volta completo.

Confronto tra le architetture

I nostri sequence diagrams, e di conseguenza il codice, sono strutturati in modo da supportare diverse funzionalità avanzate, per questo motivo risultano essere ricchi di elementi e potrebbero essere più complicati da interpretare rispetto a schemi che prevedono il supporto a meno funzionalità. Lo studio dei diagrammi dei colleghi, che non implementano le stesse funzionalità, ci ha permesso di ragionare sui blocchi essenziali e verificare che la struttura del nostro protocollo di comunicazione fosse costruita attorno agli elementi corretti.