

Peer-Review 2: Sequence Diagram

Tommaso Crippa, Alessandro Di Maria,
Mattia Brandi, Rodrigo Almandoz Franco
Gruppo IS24-AM42

7 maggio 2024

Valutazione del Sequence Diagram delle classi del gruppo IS24-AM51.

1 Lati positivi

Di seguito analizzeremo i lati positivi del Sequence Diagram

- Si può notare un'ottima distribuzione dei compiti tramite l'utilizzo di classi specifiche. Questa architettura dovrebbe rendere più agevole la fase di debug in caso di errori, poiché ad ogni classe sono assegnate poche funzionalità.
- Il documento allegato dei diagrammi è completo, infatti descrive in modo esaustivo l'interazione tra server e client.
- L'uso delle interfacce è corretto, poiché consente alle classi che le utilizzano di essere a conoscenza di quali metodi chiamare.
- Durante la fase di connessione, la logica di rete è ottimale, poiché in caso di successo invia direttamente i dati delle partite anziché richiederli nuovamente dopo aver ricevuto conferma.
- In generale, l'architettura risulta altamente scalabile, ideale per lo sviluppo del gioco richiesto.

2 Lati negativi

Di seguito analizzeremo i lati negativi del Sequence Diagram:

- Ci risulta che bisognasse fare un Sequence Diagram più astratto, valido sia per Socket che per RMI.
- Il Sequence Diagram, valido per RMI, dovrebbe avere un livello di astrazione maggiore, dal momento che l'interazione tra controller e model potrebbe non essere specificata.
- Nel Sequence Diagram è assente la fase di setup durante la quale viene inizializzata la "hand" di ciascun giocatore e vengono scelti gli obiettivi personali insieme al colore della propria pedina.
- Nella fase "Placing a Card" non viene visualizzato l'aggiornamento dei punti del player che ha piazzato la carta.

3 Confronto tra le architetture

Per quanto riguarda un confronto tra le architetture abbiamo individuato le seguenti analogie e differenze:

- Il nostro Sequence Diagram è meno distribuito, poiché l'interazione tra client e server è più compatta, mentre il Sequence Diagram analizzato ha un livello di distribuzione maggiore poiché maggiore è il numero di interfacce presenti nel network
- Un'analogia riguarda il comportamento che viene attuato quando un tentativo di connessione va a buon fine, infatti, entrambi mandiamo le informazioni riguardanti il match a cui il player si è connesso evitando di mandare un messaggio che notifica la corretta connessione
- Una differenza riguarda il modo in cui il client viene notificato della fine del game. Nella nostra logica di gioco è il server che nel messaggio di aggiornamento della view, mandato dopo un'azione compiuta, notifica anche lo stato di gioco corrente a tutti i client, mentre nel Sequence Diagram analizzato è il client a chiedere al server se il game è terminato oppure no.

- Una evidente differenza tra le architetture è il volume della logica affidata al client: il nostro è un thick client, che tramite l'utilizzo dello stato ricevuto dal server manda solo messaggi corretti. Quello analizzato invece è un thin client, senza logica di gioco, più leggero ma con la possibilità che mandi messaggi errati, aumentando il traffico di rete.