

Peer-Review 2: Sequence Diagram

Jacopo Piazzalunga Mattia Piccinato Francesco Rita
Simone Romanò
Gruppo GC26
8 Maggio 2023

Valutazione del diagramma UML delle classi del gruppo GC16.

1 Lati positivi

La struttura di rete individuata è adatta a supportare indistintamente l'utilizzo di Socket o RMI. Inoltre, questa è quasi trasparente all'implementazione distribuita del pattern MVC, conservato anche dopo l'integrazione della parte di comunicazione in rete.

2 Lati negativi

Non abbiamo individuato particolari lati negativi, in effetti anche meno che nella precedente Peer Review. Un primo dettaglio consiste nel fatto che nel Sequence Diagram illustrativo della `CREATE_REQUEST` da parte della View, sembra che il Controller non aggiunga il giocatore alla Lobby Model ma solo al LobbyController; presumiamo che si tratti di una dimenticanza perchè il resto delle richieste sembra ben formato, ma se così non fosse significherebbe che parte del Model è contenuto nel Controller, il che preclude i vantaggi del pattern MVC. Un altro aspetto non di fondamentale importanza è il fatto che gli Observer del Model (tra quelli illustrati, `NetworkLobbyObserver` e `NetworkGameObserver`), dopo essere stati notificati di un evento dal rispettivo Controller, fanno una nuova chiamata sul Model per ottenere la View, quando questa potrebbe essere passata come parametro alla notify.

Proprio perchè tali Observer ascoltano il modello per riceverne la View ad ogni aggiornamento e null'altro (perchè per altri scopi ci sono altri Observer), allora infittire lo scambio di invocazioni di chiamate complica inutilmente la logica. E, infine, l'ultimo aspetto che potrebbe essere migliorato è che il Controller non ha bisogno di contenere un riferimento al Client. Questo non comporta grandi problemi se non una perdita di generalità nel codice, che potrebbe nascondere alle classi previste dall'MVC il fatto che questo sia distribuito.

3 Confronto tra le architetture

Come nella precedente Peer Review, le rispettive progettazioni dei due gruppi ci sembrano molto simili. Tra i punti di forza del gruppo GC16 individuiamo in primis il fatto che vi è una più forte divisione tra la rete e l'MVC (nonostante il riferimento al Client nel Controller, il nostro gruppo dovrebbe ancora rivedere alcuni aspetti al riguardo); a seguire, la View viene controllata dal Controller, mentre quella prevista dal nostro gruppo sa quando iniziare a visualizzare la grafica della partita, il che implica una perdita di generalità nel caso in cui la mappa degli stati della View dovesse complicarsi; e, infine, il nostro gruppo attualmente manda sempre un oggetto Game che fa da wrapper al Model, a Chat e varie eccezioni, il che comporta un pesante scambio di informazioni superfluo che, invece, il gruppo GC16 ha evitato per mezzo di un'opportuna definizione di Eventi (Messaggi nel nostro caso).