

Peer-Review 2: UML e Rete

Enrico Tirri, Rebecca Tacchi, Giorgio Negro, Luca Prada
Gruppo 36

9 Maggio 2023

Valutazione del diagramma UML e dello strato di Rete del gruppo 26.

1 Lati positivi

1.1 Controller

Ottima scelta avere un controller unico per quanto riguarda la gestione della singola partita.

1.2 Funzione del network

Può essere una buona idea per la gestione degli errori avere un messaggio standard per ogni tipologia di errore sollevato (es: caso di azione non valida, caso di numero di giocatori inserito non valido, messaggio di riconnessione in caso di perdita di connessione).

1.2.1 Lato client

L'utilizzo di Listener permette una buona scalabilità delle componenti lato Client e in generale di componenti che necessitano di essere aggiornate di eventuali modifiche del Model.

2 Lati negativi

2.1 Model

Può essere un problema l'invio dell'intera ModelAndView: nel caso in cui venga infatti implementato un nuovo componente model che non viene aggiornato una sola volta a turno (come può essere la chat), è molto probabile che vengano inviati numerose volte le stesse informazioni attraverso la rete. Questo potrebbe influenzare la buona espandibilità del progetto.

2.2 Controller

Dovrebbe però essere implementata da un controller anche la gestione della creazione e accesso di partite multiple, in quanto trattasi di logica di funzionamento e di interazione con il Model. Tuttavia attualmente, la gestione delle partite multiple è data a gestire ad un oggetto di rete, ovvero ServerImplementation, che possiede la lista lobbyWaitingToStart.

2.3 Funzione del network

ServerImplementation rischia di rappresentare un collo di bottiglia per quanto riguarda le interazioni di rete, in quanto tale oggetto è l'unica istanza in grado di elaborare i messaggi sia di creazione e accesso alle partite ma anche di interazione di tutti i giocatori con tutte le partite.

Lo strato di rete non è ben celato ai componenti di MVC, in modo particolare per quanto riguarda la View.

3 Confronto tra le architetture

Le architetture sviluppate trovano unicamente contatto nell'utilizzo di un controller unico per la gestione di una singola partita.

Simile è l'utilizzo di un pattern a eventi per quanto riguarda la notifica di cambiamenti al model e corrispondente aggiornamento della view.

In generale non rileviamo particolari aspetti che possano migliorare la nostra architettura di Controller e Rete.