

PEER-REVIEW 2: SEQUENCE DIAGRAM

Matteo Colombi, Andrea Colombo, Leonardo Conti, Lorenzo Demontis
Gruppo AM16

8 maggio 2024

Introduzione

In questa peer review abbiamo analizzato i sequence diagram del gruppo AM25, che modellano il flow principale della partita, l'evento di disconnessione di un giocatore e l'evento di riconnessione di un giocatore dal punto di vista del protocollo di rete, ovvero l'interazione fra client e server, quest'ultimo diviso in Model, Controller e interfaccia di comunicazione (denominata *Server* nei diagrammi). Durante l'analisi dei diagrammi, abbiamo osservato alcuni lati positivi e negativi, cercando di estrapolare buone pratiche con cui migliorare il nostro modo di gestire gli eventi di rete nell'esecuzione della partita, e di aiutare i nostri colleghi a migliorare il loro.

1 Lati positivi

- i) I sequence diagram presentati sono chiari e presentano tutte le informazioni necessarie per comprendere a fondo la comunicazione che avviene tra il server e i client.
- ii) La fase di setup (scelta del lato della carta iniziale, colore, obiettivo personale) è gestita con un solo messaggio. Questo semplifica notevolmente la fase iniziale del gioco, nonostante presenti alcune criticità (vedi lati negativi).

2 Lati negativi

Osservazioni generali

- i) Dato che, come indicato nelle note che accompagnano il diagramma, tutti i messaggi (tranne quelli di errore) vengono inviati in broadcast ai client, tutti i client sono a conoscenza delle informazioni private degli altri giocatori. Ovviamente i client possono essere programmati

per non mostrare informazioni a cui l'utente non dovrebbe avere accesso, ma questa non è una soluzione che tiene conto della sicurezza del programma e sarebbe comunque da evitare: la soluzione migliore sarebbe di non inviare ai client informazioni a cui l'utente non dovrebbe avere accesso (in particolare, gli obiettivi personali degli altri giocatori e, in base alle interpretazione effettuata delle regole del gioco, anche tutte le informazioni sulla mano dei giocatori).

- ii) In alcuni casi (quando viene piazzata o pescata una carta) viene inviato agli utenti l'intero stato della `PlayArea` o della mano del giocatore che ha effettuato l'azione. Questo funziona ma ovviamente non è molto efficiente, sarebbe meglio inviare solo le informazioni sulla carta piazzata o pescata.

Osservazioni sul *main sequence diagram*

- i) Gestire la fase iniziale della partita con un unico messaggio ha due svantaggi principali: nel caso di un errore da parte dell'utente (ad esempio nella scelta di un colore già scelto da un altro giocatore), non sembra esserci un modo per far ripetere la scelta limitatamente al fattore che ha scatenato l'`InvalidStartingChoicesException`; il secondo svantaggio è che non viene seguito pedissequamente il flusso descritto dalle istruzioni del gioco, in quanto durante la scelta del lato su cui piazzare la propria carta iniziale non si dovrebbero avere informazioni circa gli obiettivi della partita.
- ii) Non è esplicitamente modellato come verrebbe gestita la richiesta di piazzare o pescare una carta da parte di un giocatore diverso da quello che sta giocando in un dato momento: questo ovviamente può di nuovo essere evitato programmando il client in modo opportuno, ma non è da considerarsi una pratica particolarmente sicura (un client modificato potrebbe mandare richieste al server nel momento sbagliato e giocare al posto del giocatore corretto?).
- iii) Nell'ultima fase del gioco non sarebbe necessario far pescare carte nuove ai giocatori. Questo non è un errore di per sé, ma potrebbe generare bug se la fine della partita è scaturita dalla fine dei mazzi. Suggeriamo di effettuare i test opportuni per verificare che questa situazione sia gestita correttamente.

Osservazioni sul *reconnection sequence diagram*

- i) Non è stato modellato il caso in cui la riconnessione avviene nel contesto della ripresa della partita a seguito di un crash del server. Immaginiamo che questo scenario sia analogo a quello modellato, con le opportune aggiunte per gestire la notifica dei client quando la partita riprende.

3 Confronto tra le architetture

Il flusso seguito dalla comunicazione del gruppo revisionato segue tendenzialmente le stesse fasi di quello sviluppato dal noi. Per questo, non ci sono differenze sostanziali riguardo come vengono gestite le varie situazioni del gioco.

Il diagramma revisionato è, per certi versi, più dettagliato rispetto al nostro: mentre noi abbiamo modellato in modo puntuale solo alcune fasi cruciali della partita (forse arrivando a un livello di dettaglio ancora maggiore), il gruppo AM25 ha modellato abbastanza dettagliatamente tutta la comunicazione. Questo aumenta il contenuto informativo del diagramma, ma ne rende meno immediata la comprensione.

Osservazioni

- i) Il messaggio 24 precede il messaggio 25, anche se logicamente dovrebbe essere il contrario. Immaginiamo che questo sia un refuso. Lo stesso si applica al messaggio 31, che dovrebbe seguire l'eccezione lanciata dal modello.
- ii) L'idea di inserire una numerazione renderebbe più chiaro il diagramma; tuttavia, la notazione puntata utilizzata per indicare "sottomessaggi" è un po' confusionaria e non è seguita in modo consistente.
- iii) Il modo in cui vengono modellate situazioni in cui al giocatore viene ripresentata una scelta finché non fa una scelta adeguata pensiamo che non sia completamente corretto; un esempio è il messaggio 3 e i suoi successivi. Sarebbe stato meglio mettere il *loop* all'esterno dell'*alt*, con la condizione di uscita direttamente nel ciclo.
- iv) Non è stato modellato un protocollo per la chat, quindi non ci è stato possibile valutare come viene gestita.
- v) Come viene lanciato il `DisconnectionEvent` nel diagramma relativo alle disconnessioni? C'è un sistema di ping/pong?