Peer review IS24-AM01

Sommario

1 Analisi

1.1 Punti di forza

- **Formato e struttura dei messaggi**
  Pensiamo che il formato JSON sia il migliore per scambiare i messaggi, sia per la sua flessibilità e facilità d'uso, e soprattutto per il supporto per la serializzazione che JSON garantisce: se si decide di implementare autonomamente la serializzazione JSON è sicuramente il formato più adatto. Anche la struttura dei messaggi è efficiente, noi stessi abbiamo dotato i messaggi di un campo per identificarli: una volta letto questo campo il controller saprà alla perfezione come gestire gli altri dati contenuti nel messaggio.
- **Ottima gestione della fase di set up e dei turni del game**
  La fase di set up e di turni del game sono state descritta in maniera precisa ed esaustiva. Tentando di simulare una partita seguendo il flusso dei diagrammi di queste fasi del gioco non si riscontra nessun problema.

1.2 Punti deboli

- **Poca chiarezza nella fase di creazione del gioco**
  Dai diagrammi di flusso della fase di user authentication e di create game and join risulta poco chiaro come un nuovo game venga creato. Leggendo i diagrammi il nostro dubbio principale è stato nella fase di authentication: non abbiamo ben compreso come sia possibile aggiungere un giocatore alla lista se il game ancora non è stato creato, poiché il game viene creato successivamente nella fase di create and join. Abbiamo pensato che una soluzione possa essere distinguere il primo login dai successivi: il primo player a fare login oltre ad inviare un nickname necessario ad autenticarsi invia anche il numero di giocatori della partita (tutti i player invieranno sia nome che numero, ma solo quello del primo player ad autenticarsi sarà preso in considerazione); in quel momento viene creato il game, e il game si pone nello status AWAITING_PLAYERS che già avete implementato.
- **Poca compattezza nei diagrammi**
  Pensiamo che gestire i casi particolari, spesso quelli in cui si lancia un'eccezione, in diagrammi separati renda i diagrammi meno compatti e di più difficile lettura. Consigliamo di implementare ogni fase di gioco in un solo diagramma, eccezioni comprese: questo rende più facile la simulazione dei messaggi scambiati durante un'esecuzione, compresi quelli scambiati quando qualcosa non ha funzionato come avrebbe dovuto.

2 Consigli

- Consigliamo di implementare il type dei messaggi JSON attraverso una enum. Avendo scelto noi stessi di dotare i messaggi di un codice che spieghi come interpretarli, la enum è la soluzione più chiara ed elegante durante l'interpretazione.
- Dal momento che avete adottato l'architettura con virtual view, vi consigliamo di notificare i cambiamenti che avvengono nel model attraverso l'invio delle diff, ovvero solo di ciò che è cambiato. Consigliamo quindi di pensare ad un metodo che presi in ingresso uno stato del gioco e una diff è in grado di ricostruire il nuovo stato del gioco. Questo renderà lo scambio dei messaggi più efficiente.

3 Resoconto finale

Nei diagrammi di flusso sono rappresentati tutti gli elementi che possono guidare i membri del progetto verso un'implementazione efficiente e funzionante. Il protocollo di rete è completo e ben strutturato, è facile simulare i messaggi scambiati durante l'esecuzione: è un lavoro completo e ben ragionato.

Ringraziamo i nostri colleghi per la puntualità nelle consegne e per i profittevoli spunti offerti dallo studio dei loro diagrammi.

Auguriamo un buon lavoro,
AM44