Peer-Review 2: Communication Protocol

Andrea Alari, Giovanni Barbiero, Chiara Bordegari Gruppo AM33

29 aprile 2022

Valutazione del protocollo di comunicazione del gruppo AM06.

1 Lati positivi

In generale la struttura del protocollo di comunicazione è ben fatta.

Abbiamo potuto notare che il modello viene aggiornato in modo incrementale nei vari client evitando l'invio di informazioni già conosciute ed alleggerendo l'uso della rete.

Pensiamo che la scelta di utilizzare il JSON sia corretta poiché permette di rimuovere la dipendenza tra client e server nell'utilizzare lo stesso linguaggio di programmazione che sarebbe necessario se fosse stato scelto di utilizzare la serializzazione di oggetti di Java.

Notiamo, inoltre, una corretta suddivisione in fasi della comunicazione in modo da ottimizzare la verifica del ricevimento di certi tipi di messaggi rispetto ad altri.

2 Lati negativi

Anche se nel complesso il protocollo di comunicazione risulta ben strutturato, abbiamo notato degli aspetti che possono essere migliorati:

• Sono sorti dei dubbi nella gestione dei messaggi di Ping e Pong. Siccome il messaggio di Ping viene inviato dal client, non ci è chiaro come il server può notare la disconnessione di tale client.

Una possibile soluzione potrebbe essere quella di prevedere nel server un timer per ogni client che si resetti quando arriva un nuovo messaggio di Ping da tale client. Inoltre abbiamo individuato anche un'altra alternativa che potrebbe velocizzare la rilevazione delle disconnessioni. Quest'ultima prevede che sia il server a mandare i messaggi di Ping, mentre il client quelli di Pong. In questo modo, non appena il server nota che un client si è disconnesso, può terminare subito la partita notificando anche gli altri giocatori.

- Crediamo che i messaggi CharacterInfoRequestMessage e CharactersDrawnMessage non siano necessari poiché potrebbero essere già inseriti nel messaggio MatchInfoMessage nella fase di Preparazione della partita.
 - Crediamo che questo sia possibile poiché le carte personaggio vengono scelte ad inizio gioco e successivamente non possono essere più modificate.
- Il messaggio WinnerMessage non sembra tener in considerazione il caso in cui la partita finisca in pareggio (ciò può accadere per esempio se, quando termina la partita in un gioco a tre giocatori, i giocatori con il minor numero di torri hanno anche lo stesso numero di professori). Invece che contenere il solo nickname di un vincitore, il messaggio dovrebbe prevedere anche il caso in cui la partita finisca in parità.

3 Confronto tra i protocolli di comunicazione

Come voi avete fatto nel vostro protocollo di comunicazione, anche noi abbiamo deciso di adottare JSON per scambiare messaggi tra client e server poiché abbiamo ritenuto importante svincolare le implementazioni rendendo così possibile la comunicazione tra server e client scritti in linguaggi differenti.

Seppur la struttura dei messaggi scambiati sia simile, abbiamo potuto notare che voi utilizzate un approccio incrementale nella gestione delle modifiche del modello. Inoltre sembra che sia presente della logica a lato client per gestire le mosse possibili in un determinato turno di un giocatore.

A differenza vostra noi abbiamo deciso di implementare un thin client senza alcuna logica al quale il server invia tutto lo stato della partita ad ogni modifica che viene effettuata e, se il suo turno, anche le possibili mosse dalle quali ne sceglie una.