

Peer-Review 2: Protocollo di rete

Moretti Lorenzo, Mormile Matteo, Orofino Antonino
Gruppo 6

1 luglio 2022

Valutazione del protocollo di comunicazione del gruppo 5.

1 Lati positivi

La presenza di lobby per la scelta della partita ci è particolarmente piaciuta, poichè non vincola nuovi player alla modalità di partita selezionata dal giocatore che si è collegato prima di lui. Anche la presenza di un messaggio esplicito di ack inserito dal gruppo 5, può portare a vantaggi nell'implementazione dell'architettura client-server

Sono stati anche individuati correttamente i casi opzionali nei quali un giocatore abbia inserito un nickname già utilizzato da qualcun altro e la richiesta di una nuova carta assistente nel caso che quella giocata non sia valida per le regole del gioco.

2 Lati negativi

2.1 Ack message

Analizzando i vari tipi di messaggi abbiamo trovato che spesso si è scelto di creare messaggi appositi di conferma invece che prediligere l'utilizzo del messaggio ack, già definito all'inizio. In particolare nel caso di messaggi di tipo `Join game`, `Start game` e `Login`, riteniamo che i relativi messaggi success siano superflui. Nella descrizione viene definito correttamente il messaggio `ack` come messaggio server - client ma crediamo ci sia un refuso, poichè nella risposta al messaggio `Join success`, l'ack viene mandato dal client e non dal server.

2.2 Carte personaggio

Altra cosa che ci sentiamo di segnalare è l'assenza dei messaggi che prevedono l'attivazione delle carte personaggio. Alcune carte, quali ad esempio la scelta dello studente da muovere dalla carta all'isola (vedi carta personaggio numero 1), prevedono la richiesta al client di una scelta.

2.3 Game object

Infine crediamo sia meglio che il messaggio `JoinGame` contenga una lista di informazioni più semplici, correttamente estratte dal model, invece che tutto l'oggetto game. La serializzazione del game object richiederebbe la serializzazione di un numero notevole di classi collegate ad esso.

3 Confronto tra le architetture

Anche se dai sequence diagram non è possibile stabilire le classi che intercorrono nella gestione della comunicazione, troviamo alcuni punti in comune nel tipo di messaggi scambiati tra server e client durante la creazione e l'adesione di un player a una partita. Infatti, se per entrambe le architetture, viene chiesto al player che vuole creare una nuova partita il numero di giocatori e la modalità, solo il gruppo 5 ha aggiunto la possibilità di poter scegliere a quale partita aderire da un elenco mostrato al player.

Al contrario di quanto visto dal gruppo 5, nel nostro protocollo le azioni vengono gestite con un unico messaggio **Action** che contiene il tipo di azione effettuata (ad esempio il movimento di una pedina dall'ingresso alla sala da pranzo) e gli attributi necessari a rappresentare il movimento. Nel nostro caso, analizzando le mosse, abbiamo individuato attributi comuni alle azioni e quindi optato per la scelta di un'unica azione, seppur l'approccio utilizzato dal gruppo 5 non presenta errori di modellizzazione o concettuali. Infine la presenza di una classe ausiliaria al game, ci permette di serializzare più facilmente il game model e rendere lo scambio di informazioni di gioco più semplice e veloce

Un'ultima differenza riguarda la presenza di interfacce strutturali per i messaggi per differenziare quelli provenienti da server a client e viceversa. In questo modo, tutti i messaggi che appartengono alla stessa tipologia implementano lo stesso metodo.