

Peer-Review 2: Protocollo di Comunicazione

Alessio Buda, Dario Mazzola, Gabriele Munafo'

Gruppo 21

Valutazione della documentazione del protocollo di comunicazione del gruppo 20.

NOTE: In diversi casi nei sequence diagram sono state omesse le risposte ai messaggi. Potrebbe comunque andare bene non specificare ogni volta i messaggi di errore o successo, dato che sono indicati nella documentazione.

Lati positivi

Osservando i sequence diagram notiamo che l'aggiornamento del model sul client avviene mediante messaggi che indicano di volta in volta le modifiche apportate senza reinviare l'intero model. Questa scelta ci sembra interessante in quanto alleggerisce il protocollo di comunicazione, permettendo un aggiornamento più immediato (es. **SetupCloudCards**, che potrebbe anche essere un messaggio utile per capire il cambio di fase di gioco).

Lati negativi

- Si tratta sicuramente di una semplice dimenticanza ma manca la descrizione del messaggio **Connect** nella sezione di descrizione dei messaggi. Risulta quindi difficile capire esattamente la sua funzione. Se esso rappresenta la creazione di una nuova socket da parte del client e la corrispondente connessione al server allora probabilmente non si tratta di un vero e proprio messaggio. Se invece rappresenta la connessione al gioco, allora è ridondante in quanto coincidente con il messaggio di **Login**.
- Il messaggio **Login** non è coerente tra documentazione e sequence diagram. Nella documentazione infatti viene specificato che il messaggio di **Login** viene utilizzato anche per determinare se la partita vada creata o meno, mentre nel sequence diagram sembrerebbe utile solo a determinare se lo username scelto sia disponibile o meno. Per risolvere il problema, consiglieremmo di porre un parametro nel messaggio di **Acknowledgement** che dica se il giocatore è il primo o meno e dunque faccia capire se serve creare una nuova partita.
- Dai sequence diagram non emerge chiaramente la possibilità, espressa nel regolamento, per il giocatore di scegliere se posizionare lo studente da spostare dalla Entrance su un'isola o nella propria dining room. Inserire un blocco alt nel diagramma potrebbe rendere più chiara questa dinamica.
- I messaggi **AddStudentsToEntrance** and **PickCloudCard** specificano due azioni che devono necessariamente avvenire in sequenza tra di loro, secondo il regolamento entrambe alla fine del turno di gioco del singolo giocatore. Potrebbe quindi essere utile avere un unico messaggio in modo da evitare possibili inconsistenze tra la scelta della cluod e il successivo spostamento degli studenti.
- Il sequence diagram della fase di pianificazione non sembra essere completa, in **playAssistantCard** non sembra esserci alcun controllo sull'utilizzo da parte di più giocatori della stessa carta (eventualmente servirebbe un loop).

- Manca il sequence diagram che descrive la fine della partita sebbene i messaggi siano descritti nel protocollo.
- Nella documentazione, probabilmente a causa di una svista, non è indicato che il messaggio di **Acknowledgement** viene mandato anche da client a server.

Confronto

- Nella nostra implementazione del protocollo cerchiamo di effettuare quanti più controlli possibili circa la validità dell'input (ad esempio numero di giocatori, che il wizard selezionato non sia stato scelto da altri giocatori, ecc.) lato client in modo ridurre al minimo lo scambio di messaggi di errore tra client e server. Questo anche attraverso l'uso di messaggi che, prima di una scelta, ricordano al giocatore le opzioni possibili (ad es. il server potrebbe inviare al client i wizard disponibili in modo che nel controller del client si possa effettuare un controllo preliminare sulla scelta del giocatore). Nel protocollo di comunicazione analizzato, invece, sembrerebbe che il messaggio di errore venga mandato dal server al client ogni volta che l'input inserito dal giocatore non risulti valido. Ci sentiamo quindi di suggerire una modifica di questo tipo.
- Nella nostra fase di login abbiamo una serie di messaggi relativi alla scelta di wizard, colore torre scelto ecc., che non vediamo specificati tra i messaggi del protocollo di comunicazione del gruppo 20. Probabilmente queste scelte verranno effettuate in una fase successiva al login, in ogni caso suggeriamo di riflettere su altri messaggi o parametri necessari alla creazione della partita. Ad esempio, nel caso in cui si voglia implementare la modalità esperto, sarebbe necessario aggiungere in **startNewGame** un ulteriore parametro per comunicarlo al server.
- Nel nostro protocollo di comunicazione l'aggiornamento delle torri sulle isole avviene in un unico messaggio, mentre il gruppo 20 ha scelto di usare due messaggi di **AddTower** e **RemoveTower**. Forse potrebbe essere utile avere un'unico messaggio di aggiornamento per tutti i client.