

Peer-Review 2: Protocollo di Comunicazione

Matteo Luppi, Paolo Lussana, Pradeeban Muralidaran
Gruppo 24

9 maggio 2022

Valutazione della documentazione del protocollo di comunicazione del gruppo 23.

1 Lati positivi

In generale la descrizione dei messaggi si presenta completa e precisa, così come i sequence diagram sono chiari e coerenti tra di loro.

Le convenzioni del linguaggio UML sono rispettate, con anche la presenza di alcuni blocchi "loop" e "opt".

2 Lati negativi

Più che lati negativi abbiamo di seguito raccolto consigli e appunti circa alcuni degli scenari analizzati e relativi messaggi utilizzati:

- **Scenario 1 (access game):** con *NewGame()* vengono passati numerosi parametri tutti insieme in un unico singolo messaggio, questo a nostro parere potrebbe rendere più difficile la procedura dei vari controlli (ad esempio che il numero di giocatori scelto sia consentito, che il nickname inserito vada bene e quindi non già in uso, e via dicendo). Così facendo, in caso di *LoadGameFailure()* risulta più complicato individuare la causa dell'errore, che sarebbe invece più facilmente rintracciabile in caso di messaggi client-server più "atomici" (uno solo per il nickname scelto, uno solo per il numero di giocatori scelto e così via).

- **Scenario 2 (planning phase):** il messaggio *PlanTurn()* risulta forse eccessivo e ridondante, in quanto c'è già un precedente ack da parte del player, il quale quindi potrebbe subito procedere a scegliere la propria carta assistente con il messaggio *ChooseCard()*.

Inoltre a nostro avviso i messaggi *ChooseCard()* e *PlayCard()* potrebbero aver più senso se separati l'un l'altro da *UniqueCard()* che dovrebbe quindi stare in mezzo, cosicché la carta assistente viene effettivamente giocata solo se viene prima superato il controllo di *UniqueCard()*.

- **Scenario 3 (action phase 1):** il messaggio *MoveStudent()* è forse fin troppo generico così, noi ad esempio abbiamo pensato a due messaggi diversi, a seconda che si vogliano spostare gli studenti o nella propria sala o su un'isola, aggiungendo inoltre come parametro il colore dello studente da spostare (e nel secondo caso l'indice dell'isola su cui piazzarlo).

3 Confronto

Nel complesso, casi sopracitati a parte, le nostre due documentazioni del protocollo di rete mostrano molte analogie negli scenari richiesti e nella sequenza di messaggi scambiati, ad esempio il nostro messaggio *RenderView()* corrisponde al vostro messaggio *GameboardUpdate()* e come questo tanti altri.