

# Peer review 2: Protocollo di comunicazione

Beghetto Pietro, de Donato Simone, Dimaglie Gregorio  
Gruppo **AM46**

Valutazione del protocollo di comunicazione del gruppo **AM19**.

## Lati positivi

- **Risposte dal server con opzioni tra cui scegliere:** inviare le possibili scelte al client interessato potrebbe ridurre il numero di errori dovuti a messaggi illegali/sbagliati da parte del client. Potrebbe anche aiutare a costruire un'interfaccia più intuitiva e più ricca di informazioni (per esempio un elenco a discesa di opzioni da cui scegliere in alcuni contesti), potenzialmente riducendo allo stesso tempo lo "sforzo" logico sul client.
- **Controllo lato server di tutti i messaggi mandati dal client:** verificare dal server che il messaggio ricevuto sia quello atteso e che le informazioni contenute nel messaggio non contengano azioni illegali ha due effetti: irrobustire il programma rendendo il server più stabile ad eventuali input mismatch (per forza di cose), ed eliminare il problema di un possibile "malicious client" diminuendo, se lo si desidera, la parte di logica che gli compete (se tanto può essere comunque manomessa...).  
L'inaffidabilità del client, pur non essendo una specifica richiesta nel progetto, è uno dei concetti fondamentali delle architetture client/server nei giochi online, e ragionare con questo presupposto è secondo noi appunto un win-win.

## Lati negativi

- **Nessun raggruppamento dei messaggi in super classi:** creare 2/3 classi padre per dividere i messaggi nelle categorie principali potrebbe essere utile per capire subito il tipo di messaggio che sta arrivando senza dover leggerne i contenuti, permettendo di distribuire la logica di interpretazione dei messaggi in un modo più chiaro.
- **A volte non vengono informati i client dopo un'azione che cambia lo stato:** Dopo alcune azioni compiute da un client (carta assistente giocata, studenti spostati) il server aggiorna lo stato e comunica gli aggiornamenti al solo al client che li ha causati, quanto in realtà le informazioni servirebbero (o comunque sarebbero utili) anche al resto dei giocatori. Un effetto di questa mancanza è la dissonanza in cui il client gioca in tempo reale, ma non vede in tempo reale il gioco degli avversari; un altro è sembra che il giocatore non venga a sapere subito che una

certa carta assistente è stata giocata, ma lo possa solo dedurre in base alle carte selezionabili fornite dal server.

## Confronto tra le architetture

- Non viene comunicato a tutti i giocatori di chi è il turno corrente, né tutte le informazioni provenienti dalle scelte degli avversari durante la fase di pianificazione. I messaggi del server sembrano essere spesso diretti solo al client che in quel momento sta inviando messaggi, mentre noi effettuiamo quasi sempre dei broadcast.
- Il server spesso chiede informazioni al client sulla prossima azione da compiere, invece che lasciare che sia il client a mandare informazioni su un'azione, e il server a *rispondere* con delle informazioni sull'esito dell'azione. Dopo esserci confrontati con più tutor, il nostro gruppo ha invece constatato che il metodo migliore per approcciare la comunicazione client/server fosse che il server, rispondendo alle azioni del client, mostrasse tutte le informazioni necessarie e selezionabili. Quando necessario quindi, il client modifica il suo contenuto mostrato in base alle risposte inviate dal server. (In questo modo il client realizza quello che nel corso di sistemi informativi è chiamato *presentation layer*: il client non prende per primo l'iniziativa ma agisce in base a quello che gli viene mostrato dal server, lo mostra, e aspetta un'interazione dell'utente, che manda senza analizzarla)
- In relazione al punto precedente, ci è sembrato estremamente rilevante che il server inviasse, prima di ricevere le azioni, tutte le opzioni disponibili in modo da eliminare quanto più possibile anche eventuali classi sul client che custodissero i dati, una specie di "copia temporanea del modello" che altrimenti sarebbe stata obbligatoria e a cui noi inizialmente avevamo pensato. Rendendo obsoleta questa struttura si alleggerisce e semplifica il client.

Grazie per l'attenzione.