

Peer-Review 2: UML Model + Controller e Protocollo di Comunicazione

Rabbolini Alessandro, Poli Matteo, Palazzi Matteo, Pirola Lorenzo
Gruppo GC02

6 maggio 2024

1 Diagramma UML

Per la sezione Model, si riveda mail precedente; sono state implementate alcune delle modifiche proposte nella precedente peer review, quali la semplificazione di getter e setter angoli nella classe Card, e sistemate alcune sviste nei metodi.

Per quanto riguarda la sezione Controller, si é lavorato a un sistema ibrido che permetta al client di selezionare a suo piacimento se utilizzare il protocollo RMI o TCP, con socket sincronizzate. Il metodo di comunicazione RMI é stato sviluppato per primo, e il funzionamento della socket TCP lo ricalca a grandi linee, permettendo facile permutazione da un protocollo all'altro. Un server TCP multithreaded (TCPSocketServer) e un server RMI sono avviati dalla classe ServerMain, contenente il metodo del controller automaticamente lanciato all'avvio del terminale come server.

2 Protocollo di comunicazione

Il server RMI utilizza, come conosciuto, l'invocazione remota di metodi come metodo di comunicazione client-server.

Per permettere a client e server TCP di comunicare, invece, é necessario un protocollo di comunicazione. Il client invia messaggi codificati come segue:

nickname#cmd#arg₁#arg₂#...

con *nickname* = nickname selezionato dal giocatore connesso al client, *cmd* = comando inviato (vd. elenco comandi), *arg_i* argomento i-esimo necessario al comando (parametro).

Il server risponde con:

Server#nickname#response#arg₁#arg₂#...

inviando la stringa "Server" di identificazione, il nickname del client a cui risponde *nickname*, una risposta *response*, ed eventuali argomenti/parametri.

NB: il server risponde anche al tentativo di connessione, per quanto esso non abbia assegnato un comando via stringa da client, per ovvi motivi.

I comandi previsti hanno una corrispondenza 1 a 1 con i metodi implementati via RMI; gli args inviati da client corrispondono alla segnatura del metodo. Il server restituisce una response di OK specifica del metodo (esempio: "drawn") o un messaggio di errore (esempio: "not_joined", "illegal_draw"), allocato dal momento in cui si denota un'eccezione e/o una return di errore nel metodo, e l'eventuale variabile di return é inviata sotto forma di args. Nel caso di argomenti array, ogni cella compone un arg separato; nel caso di argomenti sotto forma di oggetto piú complesso (esempio: oggetto Card), esso é convertito in JSON (via Gson) e cosí inviato come arg.