

Peer-Review 2: Network Protocol

Lorenzo Trombini, Niccolò Benetti, Matteo Scarlino, Salim Salici
Gruppo AM49

Valutazione del diagramma UML delle classi del gruppo **AM06**.

Lati positivi

Ci è piaciuta l'idea di condensare tutte le informazioni che il client ha bisogno di ricevere dal server in un solo messaggio (`UI Update`), pur tenendo conto del lato negativo collegato a questo che abbiamo indicato in seguito.

HandleCards State e **Draw State** sono spiegati chiaramente.

Lati negativi

Sono sempre indicati solo gli scambi di messaggi tra il server e un client, ma non è presente l'interazione con gli altri client connessi alla partita.

In fase di **SetUp** il client che crea una partita può inviare nello stesso messaggio di creazione anche il nickname che vuole usare, invece di inviarlo successivamente.

Il metodo per stabilire se ricaricare una vecchia partita (ovvero una partita interrotta da un crash del server) può portare a comportamenti non desiderati. Sarebbe meglio far inviare dal giocatore che crea la nuova partita anche un parametro che indichi l'id del vecchio game che si vuole riprendere.

Manca la parte di protocollo che illustra il caso di un client che vuole connettersi a una partita già creata da un altro utente.

Gli `UI Update` sono inviati dal server. Sarebbe invece meglio farli gestire al Client, essendo esso l'entità che sa come aggiornare l'UI in base ad aggiornamenti dello stato del model. Quindi il server dovrebbe inviare ai client solo un aggiornamento sullo stato del gioco, che poi i client processeranno e aggiorneranno la rispettiva UI in base alle informazioni contenute nell'aggiornamento.

Play state è illustrato in modo poco chiaro. Non è chiaro che informazioni siano presenti in `CornerToConnect`, `designatedCorner`, `showPossibleLinks` e `onPlay`. Inoltre non capiamo come mai venga inviato un messaggio `Return` dal controller al server prima di ogni eccezione.

In **Final State** il controller dovrebbe passare l'informazione `MapOfPlayersScores` al server (a meno che questa non sia contenuta in `onShowWinner`).

Confronto tra le architetture

A differenza dell'altro gruppo noi abbiamo preferito rendere espliciti gli ACK.

Il nostro protocollo è più granulare, ovvero abbiamo preferito rendere più esplicite tutte le varie informazioni che si scambiano client e server.