Peer-Review 2: Protocollo di comunicazione

Daniele Dente, Daniele Di Santi, Valerio Donno

Gruppo AM45

Valutazione del protocollo di comunicazione del gruppo AM18.

# Lati positivi

1. La routine di ping è estremamente utile per evitare di gestire la perdita di connessione mediante i messaggi scambiati, che così hanno unicamente funzione di comunicazione. Il timeout di 5 secondi sembra adeguato.
2. Il comando GameStatus pare un buon modo per rappresentare integralmente lo stato di gioco
3. Modello semplice e leggibile, ogni operazione è mantenuta il più atomica possibile.
4. Uso molto intelligente degli status come proxy degli elementi di gioco, che evitano di comunicare al Client informazioni non di suo interesse.

# Lati negativi

1. Riguardo la routine di ping, i messaggi scambiati da client e server dovrebbero essere invertiti, infatti nell’implementazione attuale è il server a iniziare lo scambio di messaggi, pratica inconsueta nel paradigma client-server.  
   Per risolvere questo problema è semplicemente necessario far iniziare la comunicazione (ping) al client
2. Il server invia il messaggio di inizio della partita, di nuovo l’iniziativa dovrebbe essere dei client. Più in generale gran parte dei vostri messaggi partono dal server, facendo sì che sia il client in una situazione di ascolto passivo piuttosto che di richiesta attiva.
3. Un’ulteriore conseguenza è che in ogni stato del flusso di gioco, nella variante per esperti, il server si deve aspettare come risposta sia un messaggio relativo a quella fase, sia una possibile richiesta di giocare una carta personaggio
4. La presenza del campo ServerReplyType fa sorgere il dubbio sul perché non si possa inserire un campo analogo nei messaggi inviati dal Client poiché, in una prospettiva di estendibilità, potrebbero essere accettati messaggi di natura diversa in uno stesso passaggio del protocollo.
5. In merito a quanto scritto sopra non è chiaro come venga gestito il caso di disconnessione volontaria da parte dell'utente.

# Confronto tra le architetture

1. Dovremmo implementare una routine di ping gestita su un thread parallelo, per separare il controllo di connessione su un altro thread e permettere una verifica di aliveness costante.
2. La comunicazione efficace alla View dello stato dei vari elementi del Model è un

aspetto rimasto tralasciato nel nostro protocollo. Un messaggio compatto inviato dal server che descriva totalmente lo stato di gioco è un’idea che implementeremo sicuramente. Data la relativa semplicità del modello per questa applicazione, può essere possibile rinviare integralmente lo stato dell’applicazione oltre che le sole modifiche, per semplificare il controllo a lato client e avere maggiore sicurezza sull’aderenza allo stato reale della partita.