

## Peer-Review 2: UML

Andrea Biasion Somaschini, Roberto Alessandro Bertolini,  
Gabriele Corti, Omar Chaabani  
Gruppo AM07

Maggio 2024

Valutazione dei Sequence Diagram del gruppo AM16.

## 1 Lati positivi

Per agevolare la comprensione dei lati positivi, visto e considerato l'elevato numero di diagrammi ricevuti, abbiamo deciso di schematizzare i punti per singolo diagram, come qui sotto riportato:

- **High Level - Connection Check** È evidente che sia stato prestato uno sforzo significativo per garantire una struttura chiara e una rappresentazione accurata del funzionamento del meccanismo di controllo della connessione. La comunicazione tra client e server è ben delineata. Questo fornisce una solida base per comprendere le finalità della comunicazione e il flusso delle informazioni tra le due entità.
- **High Level - Connection to Game** Apprezzabile la suddivisione accurata delle possibili azioni che un client può intraprendere quando tenta di connettersi a una lobby. Questa suddivisione facilita la comprensione delle opzioni disponibili e fornisce una panoramica esaustiva delle interazioni possibili durante il processo di connessione.
- **High Level - Game Communication** Il game loop è chiaro e ben definito, dimostrando una solida comprensione del flusso di gioco e delle interazioni tra client e server. Apprezzabile il ruolo predominante del server nella gestione della logica di gioco, preferendo un thin client.
- **Play a Card, Draw a Card** Abbiamo apprezzato l'approfondimento nelle logiche interne di comunicazione tra controller e model, seppur non richieste in un Sequence Diagram di Rete.
- **Chat** Il diagramma della chat risulta semplice e facilmente leggibile. La rappresentazione delle interazioni tra gli utenti durante la comunicazione è diretta e di facile comprensione.

Complessivamente, sebbene possano esserci delle criticità in alcuni sequence diagram, l'attenzione ai dettagli nei diagrammi valutati è evidente.

## 2 Lati negativi

Nel corso della review, abbiamo riscontrato alcune contraddizioni e ambiguità. In particolare, non è chiaro se stiamo considerando l'implementazione

della resilienza delle connessioni, poiché da due diagrammi emerge un certo dubbio. Se la resilienza fosse prevista, dovrebbe consentire ai giocatori di riconnettersi a un gioco in corso, cosa che attualmente non avviene, poiché il gioco viene terminato per tutti in caso di disconnessione (vedasi il diagramma Connection Check). Siccome avete già implementato la cosa per implementare la funzionalità di riconnessione in caso di crash del server, perchè non aggiungere anche la parte relativa alla riconnessione? Per come avete strutturato la dinamica, non dovrebbe essere troppo impegnativo.

Inoltre, il protocollo per gestire le richieste di unirsi a un gioco non risulta chiaro, siccome il client inializza la comunicazione inviando un id del game alla quale vuole unirsi, senza avere una chiara idea di quali id siano disponibili.

Un altro aspetto da considerare è l'elevato uso degli if/else, che a volte hanno reso difficile la comprensione del diagramma. Questo problema si verifica anche per la scelta dello starter side e altre opzioni, dove non è chiaro cosa accada se un giocatore non effettua una scelta. Infine, va rivista la notifica di drawCard agli altri giocatori, se il comportamento predefinito dovrebbe essere pescare una carta in base agli stati.

Riguardo ai punti specifici sollevati sui singoli diagrammi:

- **High Level - Connection Check** sarebbe più logico che il client invii un ping al server anziché viceversa, per una gestione più coerente delle comunicazioni. Il server dovrebbe semplicemente occuparsi di offrire un servizio richiesto dal client, e non dovrebbe intraprendere azioni di questo tipo di sua spontanea volontà.
- **High Level - Connection to Game** sebbene la differenziazione sia giusta, i diagrammi sono prolissi e una maggiore astrazione avrebbe reso più agevole la lettura.
- **High level - Game communication** ci sono questioni di prolissità e di notifiche ridondanti che potrebbero essere semplificate, ad esempio la possibilità di rimuovere automaticamente la carta dalla mano del giocatore quando viene chiamato il metodo placeCard.
- **Play a Card, Draw a Card** essendo un Sequence Diagram di RETE, la comunicazione tra controller e model sembra fuori luogo.
- **Chat** manca una gestione di possibili errori.

In sintesi, i diagrammi presentano varie criticità che vanno riviste per garantire una comunicazione corretta tra client e server. In particolar modo, una grave lacuna di questi diagrammi è la gestione degli errori: risulta pressochè inesistente una verifica delle informazioni inviate dai client e un messaggio di ritorno per notificare eventuali errori.

### 3 Confronto tra le architetture

Analizzando a fondo le architetture del gruppo revisionato, emerge una differenza sostanziale nell'approccio alla progettazione. La nostra struttura si avvicina maggiormente a un modello di thick client, in cui gran parte della logica di gioco è conservata dal client stesso. Al contrario, l'architettura del gruppo revisionato sembra seguire un approccio di thin client, in cui il server svolge un ruolo più pedante nel comunicare ogni singola azione che il client deve eseguire. Nel nostro caso, il client è in grado di dedurre automaticamente le azioni da intraprendere, avendo una panoramica completa del gioco. Questo approccio offre una maggiore flessibilità e autonomia al client, riducendo la dipendenza dal server per l'esecuzione delle azioni di gioco e il numero di pacchetti da scambiare attraverso il networking.