

Peer-Review 2: SEQUENCE DIAGRAM

<Luca Bigatti>, <Alessandro Bacchio>, <Alessandro Bertelli>,<Davide Ali>

Gruppo <23>

Valutazione del diagramma UML delle classi del gruppo <32>.

Lati positivi

- La struttura di rete presentata è ben progettata per la robustezza e la flessibilità del progetto, non a caso implementa facilmente tutte le varie funzionalità come resilienza alle disconnessioni, chat e partite multiple.
- Sequence diagram ben fatto e dettagliato in entrambi i casi sia RMI che SOCKET.
- La distinzione tra diversi tipi di messaggi (CtoSMessage, StoCMessage, CtoSLobbyMessage) e il loro scopo specifico (interazione con GameController o GamesManager) rende la struttura del protocollo chiara e ben definita. Le varie tipologie dei vari messaggi con i relativi campi risultano ottimi.
- Ottimo anche l'utilizzo dell'heartbeat che è stato integrato nel protocollo per garantire la resilienza alle disconnessioni dei giocatori. Questo meccanismo consente al sistema di rilevare tempestivamente le disconnessioni e di gestire in modo facile la possibile presenza di interruzioni improvvisi nella connessione dei giocatori.

Lati negativi

- La serializzazione dei messaggi può aumentare la dimensione dei dati, causando un aumento dei tempi di trasmissione.
Nella comunicazione via Socket gli oggetti devono essere serializzati prima di essere inviati sul canale, ciò potrebbe introdurre un overhead in termini di prestazioni e consumo di risorse, specialmente se gli oggetti sono complessi.
- Non è chiara l'utilità di avere una VirtualView "privata" e una AllVirtualView.
- Non è chiaro se la classe client sia in grado di astrarre la logica di rete in modo tale da potersi interfacciare al server, indipendentemente dalla tecnologia mediante la quale ci si vuole connetter.

Confronto tra le architetture

- Gestione del meccanismo di unione alla partita adottato è praticamente identico tranne per la tipologia di messaggi scambiata:
Il GameManager verifica se esiste un gioco con il nome specificato nel messaggio. Se il gioco non esiste, il GameManager invia un messaggio Upload To Client AccessGameFailed all'SkServer che manderà un messaggio di errore al client oppure esso verrà notificato tramite un accessGameConfirmationMessage se l'operazione è andata a buon fine.
- Molto simile anche la fase di creazione di un game dove anche noi utilizziamo una virtual view che notifica con un Update il client la creazione di un nuovo game, leggermente diversa risulta la modalità con cui istanziamo inizialmente le classi del controller e la stessa VirtualView che saranno utilizzate durante le interazioni client-server.