

Peer-Review 2: UML

<Spagnolo Gabriele>, <Vallone Mario>, <Zoccatelli Lorenzo>

Gruppo <GC11>

Valutazione del diagramma UML delle classi del gruppo <GC21>.

Lati positivi

1)La differenziazione di gestione tra primo messaggio (creazione lobby) e messaggi in game è apprezzata.

2)Una buona idea è stata quella di permettere agli utenti di unirsi ad una partita già creata inserendo l'ID del match (questo se abbiamo interpretato correttamente il ruolo di UUID). Probabilmente introdurremo anche noi tale funzionalità.

PS: anche se lo scopo di questa peer review è valutare il server, abbiamo apprezzato come vi siete interessati anche alla memorizzazione delle mosse degli utenti, in modo tale sia possibile annullare una mossa già inviata.

Lati negativi

1)Ogni connessione sembra passare da un'unica entità (Lobby server) che ridirige il messaggio ("sembra", in quanto non ne siamo sicuri): se davvero così fosse, a seconda dell'implementazione potrebbe fare da collo di bottiglia in termini di concorrenza.

2)L'UML è di difficile comprensione.

Confronto tra le architetture

1)La lista di tutti gli utenti connessi (in particolare WaitingLobby) è contenuta in una collezione (attributo statico) all'interno della classe LobbyServer. Per come noi abbiamo progettato il nostro software invece, abbiamo delegato la memorizzazione della lista ad un'entità esterna che memorizza tutti i giocatori connessi (nel nostro caso le loro classi Connection) e le relative RemoteView. Inoltre la nostra classe ricorda a quale match appartengono i vari giocatori.

2)Come previsto dal nostro protocollo, anche il vostro client invia al server entità di tipo Message ed attende una Response dal server. Tuttavia noi abbiamo preferito serializzare le nostre entità invece che convertirle in json.

3)I poteri speciali sono attivati da un'unica azione (PlayCharacterCard), che prende come parametro parecchie variabili (di volta in volta solo alcune di queste risulteranno importanti per l'attivazione di uno specifico personaggio). Noi abbiamo preferito dividere l'attivazione

delle carte personaggio in due fasi distinte (quindi tramite due diversi messaggi). Per prima cosa l'utente deve attivare una carta (client invia un messaggio di tipo UsePower). Dopodiché, ogni volta che l'utente utilizzerà il potere in questione (ad esempio scambiando due studenti), il client invierà un altro messaggio coerente con l'azione richiesta.