

Peer-Review 2: UML-Rete

Valentini Lorenzo, Vicedomini Andrea, Wang Toni, Vaccaro Lucrezia

Gruppo GC27

Valutazione del diagramma UML delle classi del gruppo GC17.

Lati positivi

Il progetto sembra ben strutturato e orientato a gestire correttamente la connessione tra client e server. Il protocollo è chiaro e consente di seguire agevolmente il flusso di esecuzione del gioco. Il pattern Command utilizzato per l'invio delle azioni offre una maggiore scalabilità e manutenibilità. La gestione dei comandi e degli eventi da parte di client e server è ben organizzata tramite l'utilizzo delle due code (CommandQueue e EventQueue).

Lati negativi

Potrebbe risultare utile l'implementazione di una funzione di ping per monitorare lo stato del client RMI da parte del server. In caso di caduta della connessione lato client, il server potrebbe gestire l'evento chiudendo la partita (nel caso non abbiate in programma di implementare funzioni aggiuntive sulla resilienza) o mettendo in attesa gli altri giocatori.

Confronto tra le architetture

Il Gruppo GC17 utilizza un meccanismo a doppia coda per gestire sia gli eventi dal client al server che viceversa. Sebbene il nostro progetto non lo includa ancora, la vostra spiegazione ci ha fornito uno spunto ulteriore per svilupparlo, poiché sembra molto utile per evitare blocchi nella comunicazione di rete. Entrambi i progetti implementano un minimodel nel client. Per quanto riguarda la connessione client-server, la principale differenza risiede nel fatto che la vostra gestione dei comandi avviene tramite il pattern Command, mentre noi abbiamo scelto di chiamare i metodi del server direttamente.