

Peer-Review 1: UML

Lorenzo Valentini, Andrea Vicedomini, Lucrezia Vaccaro, Toni Wang

Gruppo GC27

Valutazione del diagramma UML delle classi del gruppo GC17.

Lati positivi

L'UML del gruppo GC17 si presenta come un lavoro ben strutturato, distintamente allineato al pattern architetturale Model-View-Controller, offrendo così una visione modulare e chiara della struttura del progetto. Viene particolarmente apprezzata la gestione degli eventi, adeguatamente implementata per orchestrare la dinamica di gioco. Inoltre, l'adozione di diversi design pattern si rivela efficace nella costruzione di varie classi, come ad esempio l'utilizzo dell'Abstract Factory per la creazione dei diversi tipi di giocatori (GuiSocketPlayer, CliRmiPlayer, etc.), e l'applicazione dello State Pattern per gestire le varie fasi del gioco.

Lati negativi

Si rileva che per la classe ObjectiveCard sarebbe stato preferibile adottare lo Strategy Pattern anziché implementare diverse voci all'interno di un'enumerazione, al fine di garantire una gestione più flessibile dei diversi metodi associati. Inoltre, le GoldCard potrebbero essere considerate come sottoclassi di ResourceCard, in quanto presentano attributi comuni e rappresentano una sorta di evoluzione delle carte risorsa. L'enumerazione "Objects" risulta poco descrittiva e troppo generica, potrebbe essere opportuno sostituirla con un termine più specifico. Inoltre, la leggibilità complessiva del diagramma potrebbe essere migliorata attraverso una maggiore dettagliatezza nella descrizione delle funzionalità e delle relazioni tra le componenti, al fine di facilitare una comprensione più approfondita e rigorosa dell'esecuzione del gioco.

Confronto tra le architetture

Il diagramma UML del gruppo GC17 presenta alcuni elementi che, al momento, non sono ancora stati implementati nel nostro progetto, come ad esempio la gestione delle eccezioni, la view e la connessione tramite socket e RMI. Tra i punti di forza del loro approccio architetturale, va segnalato l'utilizzo dello State Pattern per gestire il flusso di gioco e la struttura ben definita nella gestione degli eventi. Questi aspetti potrebbero essere adottati anche nel nostro progetto, ad esempio tramite l'implementazione di una classe dedicata alla gestione degli eventi mediante listener, al fine di migliorare la struttura e la gestione del flusso di gioco.