

Peer-Review 1: UML

Airoidi, Confalonieri, Ettore

Gruppo AMo3

Valutazione del diagramma UML delle classi del gruppo AM33.

Lati positivi

Dall'analisi dell'UML si nota un buon approccio alla suddivisione del modello in sotto-package, tale approccio permette di esporre solamente i metodi strettamente necessari applicando efficacemente il paradigma dell'**information hiding**.

Abbiamo apprezzato anche l'utilizzo del pattern "**composite**" per l'estensione delle funzionalità relative alle regole base verso quelle destinate alle regole per la modalità esperta. Utile anche l'implementazione di un factory method che permette di nascondere al metodo chiamante la procedura di creazione della partita.

Lati negativi

Pensiamo che la gestione delle **isole a blocchi**, con una classe isola e una classe per il gruppo di isole, risulti complessa. Noi abbiamo invece pensato di gestire l'astrazione delle isole tramite un'unica classe "Island" che rappresenta sia isole singole che gruppi di isola, l'unificazione è gestita unendo le isole in una sola contenente tutti gli oggetti delle isole originarie.

Abbiamo inoltre notato che la scelta delle squadre è vincolata dall'ordine in cui vengono salvati i giocatori. Tenendo conto che avete una classe di tipo enumeration per **salvare i parametri** di gioco sarebbe carino utilizzarla per memorizzare le informazioni relative ai team.

Confronto tra le architetture

Confrontando il vostro ed il nostro diagramma UML abbiamo apprezzato il sistema di gestione delle **carte con effetto** che fa uso di una classe dedicata "GameModelExpert" che espone un'interfaccia tramite cui i personaggi influenzano il gioco.

Un'altra differenza con la nostra architettura è che noi abbiamo deciso di utilizzare generalmente un'**implementazione a oggetti** mentre voi avete optato per una **implementazione a contatori**, ad esempio nella gestione delle torri e degli studenti. Abbiamo scelto questa strada per avere un modello il più fedele possibile al gioco fisico, una scelta di questo tipo a noi risultava in generale più estendibile e più comoda in vista dell'implementazione di funzionalità aggiuntive come ad esempio la "persistenza".

In concreto abbiamo deciso di dividere anche il nostro modello in sotto-package e di adottare un sistema di gestione dei personaggi simile al vostro.