

Peer-Review 1: UML

Massimo Perna, Lorenzo Migliari, Alessio Migliore
Gruppo 67

4 aprile 2022

Valutazione del diagramma UML delle classi del gruppo 66.

1 Lati positivi

- **Centralizzazione e robustezza**

L'UML del gruppo 66 appare sin da subito improntato ad una implementazione MVC basata su un *model* robusto, costituito da poche classi al quali sono demandati molti compiti. Secondo noi la scelta di usare un concetto di model "centralizzato" premia la semplicità e la linearità di future implementazioni. In particolare, ci è apparsa estremamente valida la scelta di lasciare la maggior parte del controllo del gioco alle classi **Player** e **GameModel**.

- **Implementazione di Observer**

A supporto della precedente considerazione notiamo anche l'implementazione dei primi *observer* all'interno del *model* pronti a supportare una prossima implementazione della *view*.

- **Ottima scelta del pattern Strategy**

Ci appare anche estremamente valida la scelta di implementare le modalità a 2,3,4 giocatori tramite uno *Strategy Pattern* che risulta essere estremamente funzionale ai fini del riuso del codice, oltre ad essere la scelta "stilisticamente" migliore.

In generale l'UML risulta abbastanza maturo con tanto di completa integrazione delle classi necessarie per implementare la modalità per esperti.

2 Lati negativi

- **Sovraimplementazione di classi**

La scelta di implementare classi quali **Student**, **Teacher**, **Tower** ci sembra un inutile appesantimento del progetto; sono classi che devono contenere un solo valore oltre ad essere, di fatto, immobili una volta posizionati su una board o su un'isola(lavoro facilmente svolgibile da una semplice variabile **Color**), le classi inoltre non sembrano riutilizzare bene il codice in quanto potrebbero essere tutte istanziate da un eventuale classe astratta **Pedina** essendo accomunate dallo stesso metodo e attributo.

- **Mancanza di descrizione di meccaniche fondamentali**

Manca inoltre un abbozzo a meccaniche fondamentali per il gioco quali l'unione delle isole e la gestione dell'influenza. Nell'UML non risulta infatti nessun riferimento alla prima meccanica e seppur la seconda sia comprensibile non appare subito molto intuitiva all'interno del diagramma. Ci è sembrato fondamentale segnalare questo aspetto in quanto l'implementazione di tali dinamiche si è posto da subito come principale problema implementativo del nostro progetto.

- **Comprensibilità dell'UML**

Siamo infine costretti a muovere un'osservazione più stilistica che funzionale all'aspetto visivo dell'UML che appare estremamente confuso e disordinato a primo impatto.

3 Confronto tra le architetture

Indubbiamente la "centralizzazione" del gruppo 66 risulta essere l'elemento di maggior differenza rispetto al nostro progetto. Contrariamente, abbiamo deciso di optare per una distribuzione dei compiti a più classi. Questo ci ha permesso di ottenere oggetti più semplici dal punto di vista delle responsabilità/attributi ed una più efficace gestione del workflow tra i membri del team. Sicuramente la scelta dello *Strategy Pattern* per la modalità a 2,3,4 giocatori appare vincente rispetto alla nostra scelta di implementare un *Factory Pattern* che effettivamente potrebbe risultare troppo macchinosa e ripetitiva dal punto di vista del codice.