

# Peer-Review 1: UML

Angelo Attivissimo, Chiara Di Pasquale, Ginevra Bozza  
Gruppo 25

4 aprile 2022

Valutazione del diagramma UML delle classi del gruppo 55.

## 1 Lati positivi

Abbiamo apprezzato l'uso delle classi "Holder" per tenere private alcune informazioni, poichè pensiamo sia cruciale impedire l'accesso ad alcuni attributi e metodi dall'esterno, nascondendo alcune informazioni e rendendo più semplice l'implementazione.

L'UML è di facile lettura.

## 2 Lati negativi

La parte che concerne School potrebbe essere semplificata: è abbastanza complesso implementare la scuola così, invece di creare una classe per Line, Hall, TowersHolder, Tower, ProfessorRoom, Entrance, ... creare degli attributi all'interno di School con relativi metodi per gestirli più semplicemente da school, ricordatevi che alla fine Student, Professor, Tower e MotherNature sono semplici pedine che l'unica funzionalità che ricoprono all'interno del gioco è esserci o meno, quindi vi suggeriamo ad esempio per i professori di creare all'interno di School un array di booleani di 5 celle le quali rappresentano la presenza o meno di uno dei cinque professori.

Per gli assistenti invece che creare 3 classi diverse, basterebbe secondo noi

una enum contenente ognuno degli assistenti con i suoi due attributi: il valore e il movimento di madre natura. Poi creare un attributo "mano" in Wizard inizializzato con gli assistenti.

Manca una diversa inizializzazione per il differente numero di giocatori.

La classe Wizard è abbastanza ordinata, però così manca quella "cosa" che gestisce i turni, chi chiama i vari metodi come endTurn()? Wizard è una classe centrale nel UML con metodi "potenti", tipo muoveStudent, ma come si utilizza se non ha collegato la classe islandHolder, stessa cosa per chooseCloud().

Non comprendiamo bene l'utilizzo dell'interfaccia Strategy.

Secondo noi per quanto riguarda i Character conviene creare una classe astratta con soltanto l'attributo costo e il metodo effect sui quali poi verrà fatto override da ogni singola carta.

### 3 Confronto tra le architetture

Anche il nostro gruppo ha deciso di usare degli holder per fare information hiding e nascondere al controller la struttura interna del model, avevamo dimenticato però di inizializzarli nel GameBoard.

Anche il nostro gruppo ha optato per una enumerazione BLACK,WHITE,GREY per i team in quanto dà una chiara diversificazione tra i giocatori sia per il calcolo dell'influenza sia nelle partite 4 giocatori

Non abbiamo inserito la distinzione tra modalità di gioco classica e variante esperto, dovremo farlo.

Nel complesso i nostri design sono abbastanza differenti, la maggiore diversità si denota nelle classi che abbiamo utilizzato per la "gestione del turno", in quanto classi come Wizard nel nostro UML sono semplici contenitori di informazioni e non hanno grosse funzionalità nello svolgersi del turno (il nostro wizard tiene nota solo delle sue pedine).