

# Peer-Review 1: UML

Giulia Cornetta, Federica Del Beato, Samuele Delpero  
Gruppo AM51

A.A. 2021-2022

Valutazione del diagramma UML delle classi del gruppo AM24.

## 1 Lati positivi

### 1.1 Strategy

Siccome gli effetti dei personaggi sono molto pervasivi ed hanno effetto su molte classi l'utilizzo del pattern Strategy rende la struttura del codice molto scalabile in quanto uniforme nel codice la gestione di ogni personaggio. Infatti questo pattern è molto utile per modificare a runtime il comportamento di determinati metodi.

### 1.2 Interfacce

L'utilizzo delle interfacce `CanAcceptStudents`, `CanRemoveStudents`, `AcceptTower` è conforme al paradigma della Object Orientation. Queste interfacce sono utili per astrarre la proprietà di rimozione e aggiunta di studenti di più classi.

## 2 Lati negativi

### 2.1 Liste studenti

Secondo la nostra opinione la classe `GenericColored` è superflua, in quanto si limita a contenere una enum. Pensiamo che sia più opportuno utilizzare semplicemente le enum invece che creare oggetti `Tower` e `Student`. Per salvare `Tower` e `Student` nelle altre classi noi useremmo degli interi.

In relazione a questo la presenza di liste di oggetti `Student` all'interno del codice risulta un po' pesante. Ad esempio piuttosto che usare `"Map<ColorS, List<Student>"` pensiamo che sia più agevole utilizzare `"Map<ColorS, Integer>"`. Questo dovrebbe agevolare l'aggiunta e la rimozione degli studenti.

## 2.2 Mage

Il riferimento a **Mage** nella classe **Assistant** ci sembra ridondante, in quanto il **Mage** è lo stesso per tutte le carte del mazzo. Consigliamo di mantenere solo la variabile **Mage** in **Hand**.

Ulteriormente si potrebbe evitare di memorizzare la variabile **turn** in **Assistant** ma memorizzarla in **Player**. Se si decidesse di fare questa modifica si potrebbero istanziare solamente 10 assistenti, una per tipo. Le singole istanze potrebbero essere condivise dai giocatori in quanto gli oggetti della classe **Assistant** sarebbero immutabili.

## 3 Confronto tra le architetture

Rispetto al nostro design l'utilizzo del pattern **Strategy** rende la struttura più pulita.

Abbiamo notato che l'approccio usato per creare il design è molto più orientato verso la modellazione degli oggetti fisici del gioco. Noi abbiamo deciso di non avere una corrispondenza 1:1 tra gli elementi del gioco da tavolo reale e quello virtuale. Per esempio noi non abbiamo creato una classe **Mother Nature** e non abbiamo gestito gli studenti come singole istanze. La differenza tra i due non è necessariamente legato all'efficienza ma piuttosto ad una diversa concezione del design.