

# Peer-Review 1: UML

Davide Preatoni, Federico Sarrocco, Alessandro Vacca  
Gruppo GC30

3 aprile 2022

Valutazione del diagramma UML delle classi del gruppo GC40.

## 1 Lati positivi

### 1.1 Model

Le entità concrete del gioco sono state correttamente trasposte nelle loro relative classi Java, con i relativi attributi necessari. Per effettuare le principali mosse sono stati implementati i metodi necessari per modificare correttamente il model. Un altro punto di forza è l'utilizzo di strutture dati adatte al caso, ad esempio l'utilizzo di Map nelle classi *Island*, *SchoolEntrance* e *SchoolDiningHall*, che memorizzano il numero di studenti presenti in base al colore. Riteniamo che tutti i principali dati del gioco sono presenti nel modello consegnato.

### 1.2 Controller

La gestione delle singole mosse nella classe *GameController* attraverso il metodo *runCommand()* è una buona soluzione per implementare l'esecuzione della modifica dei dati del model. Una buona pratica applicata nello sviluppo è stato l'utilizzo di indici interi rispetto che a riferimenti ad oggetti come parametri nelle chiamate ai metodi.

## 2 Lati negativi

### 2.1 Model

A nostro avviso, abbiamo riscontrato solamente qualche imprecisione riguardo ad alcuni attributi pubblici nella classe *CharacterCard*, che riterremmo meglio implementati come privati con relativi metodi setter. Relativamente alle mosse, non sembrano essere presenti il movimento di madre natura e l'unione di più isole.

### 2.2 Controller

Premettendo che il modello specifica che è ancora in fase di sviluppo, non è chiara l'implementazione di come il controller modifichi il model, nello specifico del design pattern *GameStrategy*. Riteniamo il design pattern in sé corretto, ma la descrizione fornita non è stata sufficiente per capire a fondo la sua implementazione nel controller.

## 3 Confronto tra le architetture

Una differenza sostanziale tra le due architetture è la gestione dei passaggi di parametro a metodo attraverso degli indici, rispetto che a dei riferimenti ad oggetto come nel nostro attuale modello. La gestione dei comandi a livello di controller ci sembra essere un'implementazione valida, che noi non abbiamo ancora sviluppato. A differenza del gruppo GC40, il nostro gruppo non ha ancora affrontato la modellazione dell'aggiornamento della view a seguito di un evento. Per il resto, i due model sono completi e le differenze riscontrate sono principalmente di scelta di realizzazione.