

# Peer-Review 1: UML

Luppi Matteo, Lussana Paolo , Muralidaran Pradeeban  
Gruppo 24

1 aprile 2022

Valutazione del diagramma UML delle classi del gruppo 23.

## 1 Lati positivi

- in generale il diagramma UML si presenta completo, chiaro, preciso e ordinato, garantendo una buona comprensione e rispettando in pieno le linee guida dell'approccio object-oriented
- la gestione delle 12 carte personaggio risulta già più che dettagliata e rigorosa, con i metodi specifici per ogni singola carta
- il diagramma è stato ideato già tenendo conto di alcune funzionalità aggiuntive, ad esempio l'implementazione di tutte le 12 carte personaggio e la possibilità di giocare in 4 giocatori (con relativa gestione della chat abbozzata)

## 2 Lati negativi

- classi *Student* e *Professor* senza metodi e conseguente gestione degli stessi con ArrayList, potrebbero essere più semplici da gestire con dei contatori per ogni colore, o con dei flag nel caso dei professori, che indicano se un certo player è o meno in possesso di un determinato professore

- potrebbe tornare utile un attributo che funziona da flag booleano (o qualcos'altro di equivalente) in *IslandTile* per indicare la presenza di tessere divieto *noEntryTiles* su una determinata isola
- manca un attributo in *IslandTile* (o comunque un modo alternativo) per tenere traccia degli studenti presenti su un'isola, che servono per il calcolo dell'influenza
- manca forse un modo per capire quante torri sono presenti su un singolo arcipelago (o in maniera equivalente da quante isole è composto un arcipelago) per il calcolo dell'influenza e un modo per determinare quanti arcipelaghi ci sono, per decretare la fine della partita in caso questo numero sia pari (o minore) di 3

### 3 Confronto tra le architetture

La struttura generale del model, le classi che lo compongono e le relative associazioni sono a grandi linee simili alla nostra idea, se non per qualche differenza tra le quali annoveriamo:

- come sopracitato la gestione degli studenti e dei professori, e di conseguenza la gestione della *SchoolBoard*
- la gestione di carte assistenti e carte personaggio, qui entrambe figlie della classe astratta *Card*

Notevole infine la struttura dei package *View*, *Controller*, *Network* e *Event* i quali, a differenza dei nostri intanto solamente abbozzati, presentano indubbiamente già un buon livello di dettaglio.