

# Peer-Review 1: UML

RONCHIERI, PROVERBIO, PAZIENZA, TAGAB

Gruppo GC38

Valutazione del diagramma UML delle classi del gruppo GC28

## Lati positivi

Abbiamo in analisi un UML molto ben fatto e organizzato in cui vengono già gestiti nella sua prima versione molte casistiche, tra cui le eccezioni. Si tratta di un modello molto strutturato e modulare, orientato per coprire delle Funzioni Avanzate, dal nostro punto di vista ottimo.

L'utilizzo degli oggetti è impeccabile essendo una modellazione prettamente Object Oriented, poche cose sono lasciate al caso e tutto sembra già orientato per una gestione grafica: definizione di ranking, utilizzo di `TilePositionBoard` e di una `ArrayList` che permette di gestire tutto il gioco senza dover scorrere ogni volta una matrice che avrebbe molte posizioni vuote.

La gestione della partita identificata dallo Stato della stessa è un'idea che risulta essere molto comoda nel pre-game, *nel caso in cui ci siano istanziate più partite contemporaneamente*.

Il gruppo ha già chiaro gli obiettivi da raggiungere, tutti gli oggetti interagiscono in maniera ottimale.

## Lati negativi

Ambiguità nel metodo `MyShelfie` per inizializzare più partite, ad esempio, si vuole passare come parametro `"intPlayerNumber"` che rappresenta la posizione del player in un gioco e non il numero della partita stessa.

Descrizione dei `PersonalGoal` lasciata ambigua, la enum di `GoalP1` e `GoalP2` non sufficientemente dettagliata, non è molto chiaro l'utilizzo dei metodi e la definizione delle stesse. Così come in `Board`, per i tre costruttori definiti non è intuitivo capire come e dove vengono utilizzati i tre metodi (costruttori), così come i due metodi successivi che sembrano ridondati considerando solo UML del model (essendo descritti allo stesso modo). Discrepanze nelle classi tra UML e documentazione dello stesso, mancanza di: `CommonCardShape` (CCS) – `CommonCardPosition` (CCP), da nostra interpretazione: CCS descrive l'obiettivo mentre CCP va a fare l'effettivo controllo. Ma non viene specificato con precisione dove.

## Confronto tra le architetture

Rispetto al nostro modello c'è un utilizzo più appropriato delle strutture dati come gli `ArrayList` che permettono una gestione più efficiente e meno dispendiosa, rispetto a una semplice matrice. La gestione dei `Personal Goal` & `Personal Deck` ci è stata molto utile per capire l'implementazione ottimale delle stesse, avendole realizzate solo in maniera randomizzata.

Generalmente pensata con obiettivi diversi tra i due gruppi, ma, vedere un punto di vista più

completo e maggiormente ad oggetti ci è stato molto utile per apportare delle modifiche, per quanto possibile, al nostro modello.