

# Peer-Review 1: UML

Pressiani Giuditta, Rossi Gabriele, Spinelli Davide, Travaini Alessandro

Gruppo GC-19

Valutazione del diagramma UML delle classi del gruppo GC-09.

## Lati positivi

Il numero di classi è contenuto e ciò rende più facile l'accesso ai dati da parte del controller; questo aspetto è visibile simulando a mano il flusso di gioco e notando che in generale si accede al massimo a due classi per compiere un'azione.

Rendere la funzione di obiettivo comune un Predicate rende più flessibile il codice per possibili futuri aggiornamenti. Sempre riguardo le carte obiettivo comune, la loro gestione tramite un factory pattern ne semplifica l'aggiornamento delle esistenti e l'aggiunta di nuove.

## Lati negativi

Nel complesso non si sono riscontrati grossi problemi, ad eccezione di qualche sottigliezza che è doveroso far presente.

Negli obiettivi comuni, la rappresentazione dei punteggi può essere effettuata tramite l'utilizzo di un array e non di una matrice; poiché ogni commonGoalCard tiene conto dell'ordine dei giocatori che hanno completato quell'obiettivo, si potrebbe sostituire la lista di OrderOfCompletion con una mappa con il nome del giocatore come chiave e il punteggio assegnato come valore.

Nella classe LivingRoom il metodo fillBoardIfNeeded risulta superfluo vista la presenza di needRefill e fillBoard: immaginando uno scenario di ripristino del soggiorno, needRefill va a "true" e subito dopo verrebbe invocato il metodo fillBoard che metterebbe le tessere ove necessario.

Riguardo Tiles, la classe Orientation non serve e rende più complesso il modello: nonostante nella realtà le carte possano essere disposte in modo disordinato nelle quattro direzioni, questo non influisce sullo svolgimento del gioco e sul calcolo dei punti.

Il Player dovrebbe essere collegato a Bookshelf tramite una composizione, visto che la libreria non può esistere senza giocatore e viceversa.

## Confronto tra le architetture

Mettendo a confronto i due modelli, i metodi che accettano una lista di coordinate per prendere gli oggetti dal salotto e metterli nella libreria, sono migliori dei nostri che accettano una coordinata alla volta.

L'utilizzo di file Json per la funzione avanzata di persistenza è efficiente e pratico.

Riprendendo quanto già discusso nei lati positivi, la visione del gioco del gruppo GC-09 risulta più semplificata della nostra portando a un modello con meno classi e di più facile implementazione.

La maggiore complessità della nostra architettura semplifica tuttavia alcune azioni nello svolgimento del gioco, come ad esempio il riempimento delle caselle a seguito del ripristino del soggiorno.

Infine, confrontando la libreria tra i due modelli si nota che la rappresentazione del gruppo GC-09 risulta migliore perché ignora la singola "cella" della libreria, che invece noi abbiamo simulato.