

Peer-Review 1: UML

Mario Pesca, Salvatore Tuand, Lorenzo Simone, Xuwen Ye
Gruppo 24

3 aprile 2024

Valutazione del diagramma UML delle classi del **Gruppo 14**.

1 Lati positivi

- Buona idea quella di creare la classe Lobby per fare in modo di gestire le varie partite a cui si può partecipare ed eventualmente crearne una nuova.
- Viene correttamente utilizzato il design pattern Strategy.
- Troviamo la classe Globals una valida aggiunta per gestire le dimensioni immutabili del gioco.
- La classe PlayerTable "libera" la classe Player di alcune responsabilità, andando ad impattare positivamente sulla leggibilità grazie alla separazione delle informazioni.

2 Lati negativi

- Il metodo *draw()* dovrebbe ritornare una Card, quindi non dovrebbe essere void.
- Troviamo che utilizzare due matrici differenti per rappresentare il PlayerTable non sia ottimale in quanto diventerebbe necessario aggiornare le entrambe matrici ad ogni modifica per mantenerle sincronizzate, aumentando la probabilità della presenza bug.
- Non vi è nessun attributo che tenga conto di chi sia il primo giocatore, utile per sapere quando terminare la partita. (Assumiamo che il gruppo revisionato abbia deciso di tenere nota dell'ordine di gioco all'interno dell'array `matchPlayers[]`, ma questa cosa non risulta esplicitata).
- Non viene reso possibile identificare la posizione dei Corner all'interno della Card (con le stesse considerazioni del punto precedente).
- Non crediamo corretto inserire l'attributo `points` nella classe astratta Card in quanto `StartingCard` (sottoclasse di Card) non ha i punti.
- Non ci risulta intuitiva la comprensione nel nome Corner per l'enumerazione delle risorse e degli oggetti, problema che poi si riscontra con l'attributo `requirements: Corner[]` in `GoldCard`.

- Nell'enumeration Corner, sarebbe più opportuno cambiare il nome VISIBLE in quanto si è assunto che un angolo con una risorsa/oggetto sia già VISIBLE di default.
- Nella classe Lobby, troviamo che definire già gli attributi per rappresentare i vari deck non sia particolarmente utile, in quanto possono essere inizializzati direttamente nelle fasi iniziali del match, assumendo che i deck non siano stati già mescolati in Lobby, altrimenti si distribuirebbero tutti deck uguali ai i diversi match.

3 Confronto tra le architetture

Differenze con il nostro Model che verranno tenute in considerazione nello sviluppo futuro:

- La classe Lobby del model del gruppo revisionato è utile per gestire la lista di partite a cui è possibile partecipare o eventualmente crearne di nuove.
- Potrebbe essere un'idea valida per il nostro gruppo quella di adottare il pattern Strategy anche per le GoldCard.