

Peer-Review 1: UML

Marco Baratto, Stefano Bonfanti, Hanna Chyzheuskaya
Gruppo AM24

1 aprile 2022

Valutazione del diagramma UML delle classi del gruppo 54.

1 Lati positivi

- Il Pattern Factory risulta molto comodo per generare istanze diverse della classe Personality a seconda del Personaggio richiesto.
- La suddivisione in classi e il loro collegamento è coerente, gli oggetti che appartengono ad ogni singolo player (Hand e gameBoard) non sono contenuti in altri contesti.
- Utilizzare oggetti Isola che possono rappresentare sia Isole singole che aggregati semplifica le operazioni da svolgere per sostituire le Torri e calcolare l'influenza.

2 Lati negativi

- La classe Game svolge gran parte della logica mentre le rimanenti classi agiscono principalmente da contenitori di informazioni, fornendole gli oggetti tramite metodi getter e permettendole di aggiornare i valori tramite metodi setter, sarebbe preferibile delegare il lavoro agli oggetti coinvolti per non esporre gli oggetti.
- MoveStudents sembra definire solo gli spostamenti da GameBoard a Isola, mentre per gli spostamenti all'interno della GameBoard o da

nuvola a GameBoard non sono presenti dei metodi diretti che danno questa possibilità.

- Da come abbiamo compreso l'implementazione delle sottoclassi di Personality ci sembrerebbe che i metodi choose non possano restituire i valori selezionati dall'utente non avendo riferimenti ad altri oggetti o parametri in ingresso.
- Ogni volta che un Professore cambia proprietario andrà modificato un attributo sia nella classe Game che nella GameBoard di ogni giocatore coinvolto.
- Hand, GameBoard e Card contengono un attributo owner che non risulta necessario essendo unicamente accessibili dal Player che contiene tali Classi.

3 Confronto tra le architetture

- Il nostro gruppo aveva pensato di modellare molti degli aspetti fisici del gioco come gli Studenti sotto forma di oggetti. Questa implementazione invece non modella gli Studenti ma ne tiene traccia tramite il Color. Abbiamo deciso di adottare la rappresentazione per Colore in modo da evitare di creare oggetti contenenti un solo attributo.
- Nella nostra implementazione i Professori non sono memorizzati nella plancia di ogni singolo giocatore. L'unica differenza tra le implementazioni sono le operazioni da compiere quando avviene un cambiamento di proprietario.
- La nostra scelta per rappresentare i diversi personaggi è stata quella di creare più classi, aggregando in base alla funzionalità degli effetti con caratteristiche simili. La differenza tra le implementazioni consiste nel dover creare un numero diverso di oggetti Personaggio con attributi e metodi più o meno specifici.