

Peer-Review 1: UML

Karanbir Singh, Kevin Thomaj, Gabriele Tonello, Lara Tongiorgi

Gruppo GC26

Valutazione del diagramma UML delle classi del gruppo GC16.

Lati positivi

La progettazione generale è solida e sintetica, con una buona suddivisione delle classi e dei package. In particolare, la gerarchia delle classi Card modella bene le caratteristiche dei diversi tipi di carta, con una suddivisione progressiva in sottoclassi permettendo di gestire in modo efficiente le proprietà condivise. La scelta di usare un'ulteriore classe Side è altrettanto appropriata, in quanto rispecchia il fatto che una carta assume proprietà diverse in base al lato giocato, e mantiene una gerarchia essenziale, arricchendo il FrontSide con sottoclassi necessarie, ma lasciando il BackSide generico.

Inoltre, lo strategy pattern ScoreCalculator per calcolare il conferimento dei punteggi è una scelta altrettanto appropriata, e sembrerebbe essere stato implementato in modo elegante grazie all'uso di un'unica interfaccia utilizzabile sia dalle GoldCard che dalle ObjectiveCard.

Lati negativi

Relativamente all'UML:

Da un punto di vista di presentazione del progetto, l'UML presenta alcuni difetti formali. In primo luogo, gli incroci tra le linee del diagramma rendono la lettura difficile e fanno sembrare complessa ed intricata una struttura che invece sembrerebbe essere logicamente pulita. Inoltre, alcuni collegamenti non sono corretti:

I collegamenti verso l'interfaccia ScoreCalculator dovrebbero essere frecce per indicare l'utilizzo, mentre al momento indicano una relazione binaria;

Avendo messo l'attributo ScoreType in FrontSide, quest'ultimo dovrebbe essere collegato a ScoreCalculator, mentre al momento è collegato solo a GoldFrontSide;

Gli attuali collegamenti della classe Deck alle implementazioni di Card indicano che ogni istanza di Deck contiene 3 array (un array di CommonCard, un array di ObjectiveCard e un array di StarterCard), mentre formalmente dovrebbe esserci un unico collegamento all'interfaccia Card.

Infine, forse è un errore di distrazione ma il Game attualmente ha i riferimenti di soli tre Deck, mentre, nel caso non fosse un errore, la scelta di usare un unico Deck per le CommonCard non sembra essere ottimale.

Relativamente alle scelte di progetto:

La gestione dei corners in Side utilizzando tre matrici è a nostro parere l'elemento più problematico e poco elegante del progetto: è un sistema macchinoso e ridondante, che rappresenta in modo sparso le proprietà di una stessa entità/classe; inoltre, deduciamo che sia il motivo per cui nel metodo getCorners risulti necessario restituire la descrizione dell'angolo in una stringa. Alternativamente, potreste optare per utilizzare un'unica matrice rappresentando in modo sintetico le caratteristiche dei corner o con un'enumerazione più estesa, oppure creando una classe Corner che contenga le proprietà come attributi

(dunque la matrice sarebbe di oggetti Corner e potrebbe essere necessario rivedere anche la gestione dell'enum).

Un altro aspetto migliorabile è l'implementazione dei counter all'interno di Player: per ridurre il numero di attributi e facilitarne la gestione potreste utilizzare delle Map<Kingdom, Integer> e Map<Item, Integer>. L'uso di una Map potrebbe essere funzionale anche all'interno di GoldFrontSide per rappresentare i requirements, perché per come è gestito attualmente sembrerebbe essere necessario contare i diversi tipi di risorsa necessari scorrendo l'array al momento dell'utilizzo, mentre è possibile avere i dati già disponibili.

Nella chat, la mancanza del destinatario in Message non permette di inviare messaggi ad un singolo giocatore, come invece richiesto da specifica.

La classe BackSide è erroneamente rappresentata come astratta, ma potrebbe essere solo un errore di formattazione.

Infine, consideriamo che potrebbe essere utile modificare il metodo Player.finalAction in modo tale che non richieda come parametri tutte le liste da cui il giocatore può pescare la carta, ma solo le liste necessarie (es. se la carta da pescata è tra le carte risorsa scoperte, allora il metodo necessiterà della lista delle carte risorsa scoperte e della lista delle carte del deck per rimpiazzare)

Confronto tra le architetture

Valuteremo una revisione del nostro strategy pattern in modo da gestire in maniera più pulita il conferimento dei punti, utilizzando un'interfaccia apposita come adottata nel vostro modello di gioco.