Peer-Review 1: UML

Cristiana Calvaresi, Valerio Capo, Giovanni Mattia Codemo
Gruppo AM12

4 aprile 2022

Valutazione del diagramma UML delle classi del gruppo AM42.

1 Lati positivi

I componenti quali Bag, Clouds, IslandStudents e DashBoardStudents sono stati resi come interfacce implementate da un'unica struttura, la StudentContainer. L'istanziazione di uno di questi componenti come StudentContainer garantisce accesso limitato ai soli metodi definiti nell'interfaccia, ed impedisce l'utilizzo di metodi non pertinenti ai fini indispensabili del singolo oggetto.

2 Lati negativi

Risulta poco chiaro come può essere invocato il metodo mergeIslands, contenuto in IslandPanel, dichiarato come privato, nel caso in cui fosse attivato l'effetto specifico della carta personaggio che permette di calcolare l'influenza su di un'isola qualunque.
L'implementazione attuale delle tessere no entry nella classe Island non sembra permettere l'aggiunta di più di una carta no entry, in quanto al momento sfrutta un boolean.
Non ci sono riferimenti in merito alla difficoltà di gioco della partita.
É stata omesso nell'interfaccia della Bag il metodo add. Considerando che ai fini di setup del tavolo la Bag deve essere riempita due volte non è chiaro se quest'azione venga svolta all'interno del costruttore e pertanto venga creata una Bag temporanea esclusivamente per riempire le isole.

Il metodo checkProfessors in DashBoardPanel non sembra in grado di funzionare in maniera diversa nel caso la carta personaggio che permette di ottenere il professore in caso di parità di punteggio sia attiva.
Sul tavolo dovrebbe essere disponibile per specifica un numero limitato di coin, che però non sembra essere considerato nell'UML attuale.
Non abbiamo trovato dei metodi in grado di far arrestare il modello raggiunta una condizione di fine partita.
I metodi del Model che prendono un player enum in input lo potrebbero omettere in quanto sono metodi che possono essere chiamati soltanto in riferimento al player attuale. Se sono stati messi per verificare che sia il giocatore corretto ad effettuare la chiamata, si potrebbe mettere un check adeguato in un metodo apposta.
Durante il calcolo dell'influenza la Board non ha accesso alla distribuzione dei team per essere effettuato correttamente nel caso di una partita con quattro giocatori.
Alla fine di una fase d'azione allo stato attuale non sono presenti metodi per sistemare lo stato del tavolo per il prossimo round.

3 Confronto tra le architetture

Rispetto alla nostra architettura si evidenzia una distinzione tra lo stato fisico della partita e quello temporale, e ciò rende le classi più leggibili e più specializzate. D'altra parte noi abbiamo implementato alcune classi in più, tra cui la classe Student, che rendono più semplici azioni di debugging e check di input in arrivo da parte del controller (almeno nel nostro piano di gestione degli input). Tutto sommato i nostri class diagram sono simili, eccetto alcune scelte implementative sopracitate.