

Peer-Review 1: UML

Federica Tommasini, Angelo Zagami, Giuseppe Bonanno
Gruppo 18

5 aprile 2022

Valutazione del diagramma UML delle classi del gruppo 17.

1 Lati positivi

Le relazioni e le rispettive cardinalità sono definite in modo corretto.

L'idea di interfacciare il controller con la classe "Player" del model può risultare vantaggioso quando si effettuano i controlli sullo stato del gioco.

L'implementazione dell'interfaccia "StartingMenu" che permette di definire le impostazioni della partita è una buona idea.

2 Lati negativi

L'UML risulta difficile da comprendere a causa della disposizione confusionaria delle classi, ad esempio, le classi che rappresentano il model e il controller non sono raggruppate in package diversi.

La sintassi utilizzata per definire gli attributi all'interno di alcune classi non risulta corretta, inoltre, molte classi presentano tutti i metodi privati e ciò non ne permette l'utilizzo. In aggiunta, i nomi di alcuni metodi non sono esplicativi rispetto al loro utilizzo, ad esempio il metodo "drawStudent" e molti valori di ritorno non sono corretti, ad esempio "getter(): void".

Le classi "Bag" e "MotherNature" avendo pochi metodi e attributi e rappresentando un'unica istanza per ciascuna partita potrebbero essere considerate come attributi della classe "Board". Inoltre, anche per quanto riguarda la classe "Student" potrebbe essere omessa, in quanto presenta soltanto un

attributo ed è necessario istanziarla un numero elevato di volte. La classe "Island" presenta una ridondanza nell'attributo "motherNature", in quanto la posizione di madre natura può essere ricavata dall'attributo della classe "Board". Inoltre, il metodo che calcola l'influenza di un giocatore sull'isola non può accedere a nessuna informazione riguardo al numero di studenti per ciascun colore presenti sull'isola, poiché la classe ha solo gli attributi "numberOfStudentsG1" e "numberOfStudentsG2" che si suppone rappresentino il numero di studenti controllati da due diversi giocatori, se così fosse, questa implementazione non consente di tenere traccia degli spostamenti dei professori da un giocatore ad un altro.

3 Confronto tra le architetture

Una possibile modifica che potrebbe essere implementata nella nostra architettura è di utilizzare il controllo sul singolo giocatore per definire lo stato del gioco, mentre, nel nostro caso, è stata utilizzata una classe che rappresenta lo stato attuale completo del gioco.

Un ulteriore differenza è la scelta di avere due classi distinte per il giocatore e per la dashboard, rispetto alla nostra implementazione che prevede l'utilizzo di un'unica classe, potrebbe risultare vantaggioso evitando l'eccessiva complessità della classe stessa.

Interessante l'idea di avere un'interfaccia per i metodi di gestione del menù, aspetto da noi al momento non considerato.