

Peer Review 1: UML

Michael Vitali, Enrico Simionato, Manuel Tacca

Gruppo 48

Valutazione del diagramma UML del gruppo 58

Aspetti Positivi

- Pensiamo che la gestione dei Characters - nonostante l'implementazione non sia stata ancora sviluppata - attraverso l'utilizzo dei patterns sia un buon metodo per gestire correttamente l'attivazione delle carte. L'implementazione mediante Strategy Pattern sembra un buon modo per aumentare la scalabilità: sarà infatti facile aggiungere nuovi personaggi. Tutto ciò è subordinato al buon utilizzo del pattern, che è solo abbozzato nel diagramma consegnato.
- La gestione delle torri mediante un unico intero ci sembra una buona idea in quanto facilita il loro utilizzo da parte della GameBoard, inoltre facilita lo spostamento delle torri sulle isole in quanto non è necessario passare degli oggetti tramite le chiamate a metodo.
- L'utilizzo delle costanti ci è sembrato un buon modo per rendere il codice adatto al riuso. Infatti, nel caso di future modifiche nel gioco, queste costanti possono essere nuovamente settate in maniera globale, così da risultare visibile da parte dei metodi.

Aspetti Negativi

- A nostro avviso la gestione degli studenti potrebbe essere implementata in maniera più semplice costruendo una classe singleton Bag separata. Riteniamo infatti che creare un'aggregazione su GameBoard sia di difficile utilizzo; ad esempio, risulterebbe difficoltosa l'aggiunta di studenti estratti casualmente sulle carte personaggio.
- Riteniamo che la gestione generale della School sia troppo specializzata e suddivisa, in particolare la classe TowerRoom potrebbe essere gestita direttamente nella School. Lo stesso potrebbe valere per la classe Hall, che può essere passibile di una più semplice implementazione se gestita all'interno della School.
- La gestione degli assistenti giocati in un turno potrebbe essere gestita esclusivamente da Game, così da poter effettuare in maniera più centralizzata i controlli sulla possibilità di giocare o meno un determinato Assistant.

Confronto tra le architetture

Confrontando il nostro diagramma rispetto a quello del gruppo 58, abbiamo notato delle somiglianze riguardo all'implementazione della logica sulla classe Game, ipotizziamo infatti che il funzionamento logico sarà analogo al nostro, ovvero la classe Game riceverà le chiamate dall'esterno, risultando come unico ingresso tra il Model ed il Controller. Abbiamo pensato in modo simile anche la gestione dell'unione delle isole tramite una classe MergedIsland e la gestione dei Player come un'entità.

Come già evidenziato in precedenza, la nostra gestione della schoolBoard è meno suddivisa; inoltre, la nostra gestione degli studenti rimasti da pescare è regolata da una classe Bag, come Singleton, richiamabile più agevolmente in tutte le parti del codice.

Valuteremo anche la sostituzione degli oggetti Tower con degli interi, al fine di alleggerire i metodi relativi al posizionamento delle torri sulle isole e alla loro restituzione alle SchoolBoard.