Peer-Review 1: UML

Endi Kucuku, Giovan Battista Landolina, Alen Kaja - Gruppo [49]

Valutazione del diagramma UML delle classi del gruppo [48]

Lati positivi

- Organizzazione delle classi: il diagramma del modello risulta abbastanza ordinato, essendoci numerose classi che offrono modularità del codice
- Sembra esserci una separazione netta tra il model e il controller visto che il model contiene solamente metodi getter e setter e in genere non molto complessi, che svolgono compiti basilari.

Lati negativi

- Il controller potrebbe risultare verboso ed eccessivamente complicato poiché deve gestire la logica anche a basso livello essendo il model molto snello
- Manca una organizzazione generale delle classi che sembrano disunite e non è chiaro il modo in cui potrebbero essere collegate:
    - per decidere quale giocatore ottiene l'influenza su una determinata isola ci sarebbe bisogno di un metodo generale che itera sui giocatori della partita. Tuttavia non abbiamo modo di risalire a questi ultimi (manca una struttura dati che li contenga, che dovrà essere costruita nel controller);
    - non si vedono alcuni collegamenti fondamentali come ad esempio il sacchetto che dovrebbe avere un riferimento nella classe GamePanel: farebbero comodo metodi che spostano studenti dal sacchetto agli ingressi delle board o alle nuvole;
    - La classe characterCard e le sue derivate non hanno riferimenti ad alcun altro oggetto del model, per cui non possono effettuare alcuna modifica.

- Classi / metodi ridondanti o superflui:
    - non è ben chiara la classe student che non risulta implementata, essendo sufficiente associare ad uno studente un colore ed evitare la gestione con una classe, che potrebbe risultare difficile da gestire

Confronto tra le architetture

- In confronto al nostro progetto, il gruppo 48, non implementa logica di base all’interno del model, il che potrebbe tradursi in un controller troppo complesso e difficile da debuggare.
- L’uml del gruppo 48 sembra essere più ordinato del nostro e presenta meno dipendenze tra classi. Potrebbe essere più opportuno rispetto la nostro, che risulta più interconnesso.
- Dovremmo valutare se è possibile rendere tutte le carte effetto sottoclassi di una carta generica in quanto potrebbe portare semplificazioni del codice.