

Peer-Review 1: UML

Andrea Bricchi, Eliahu Itamar Cohen, Simone Controguerra, Giovanni Di Lorenzo

Gruppo AM23

Valutazione del diagramma UML delle classi del gruppo AM32.

Lati positivi

Abbiamo apprezzato l'implementazione di una classe Game che si occupa di gestire la partita nelle varie fasi di esecuzione, dotato di metodi funzionali alla programmazione ad oggetti come l'overloading del metodo CollectTile.

Ottima idea quella di aver scelto il design pattern Strategy per l'implementazione delle carte Obiettivo comune che può permettervi di selezionare il metodo corretto dipendentemente dal tipo di carta estratto.

In linea generale, tutte le classi sono totalmente complete e non avrebbero bisogno di modifiche eccezionali.

Lati negativi

Abbiamo notato che alcuni metodi potrebbero essere accorpati tra di loro poiché svolgono funzioni complementari ed unicamente significative a loro stesse, per esempio i metodi freeTiles e maxInsert.

Vi invitiamo a pensare di sostituire l'heap per la generazione delle tessere con un'altra classe rappresentante il sacchetto che ha la funzione di generare ed estrarre casualmente le tessere da posizionare sulla plancia di gioco.

Avete realizzato per le carte Obiettivo comune 12 sottoclassi, una per ciascun tipo disponibile nel gioco. A lezione ci hanno suggerito più volte di trovare degli elementi in comune per raggruppare le carte in meno sottoclassi e siamo riusciti a suddividerle in 7 classi.

Confronto tra le architetture

In conclusione, a confronto le nostre architetture sono molto simili se non per quelle differenze implementative di cui abbiamo accennato prima. Eravamo in dubbio se scegliere come design pattern la Factory o la Strategy e dopo questo confronto pensiamo anche noi di adottare lo Strategy.