Peer-Review 1: UML

Daniele Dente, Daniele Di Santi, Valerio Donno

Gruppo AM45

Valutazione del diagramma UML delle classi del gruppo AM18.

# Lati positivi

Si nota una buona astrazione dei metodi che gestiscono i comportamenti della partita, e un utilizzo di strutture dati appropriate.

Gli handler dei vari elementi di gioco offrono una buona divisione dei compiti.

Inoltre, l’interfaccia StudentHolder modella perfettamente molte delle entità presenti nel gioco, permettendo un riutilizzo molto efficiente del codice.

La gestione delle fasi è ben delineata, e il controllo di fine partita nel caso 3 alla fine della fase azione è molto naturale.

# Lati negativi

Vi sono attributi e metodi che hanno utilità solo nel caso di un gioco svolto con le regole avanzate, mentre rimangono inutilizzati se viene giocata una partita standard, delle sottoclassi potrebbero migliorare questa divisione.

L’utilizzo di puntatori null come metodo di segnalazione di condizioni particolari potrebbe essere difficilmente comprensibile, si potrebbe utilizzare il meccanismo delle eccezioni.

L’interfaccia ModifiableHandler non è molto chiara.

Nella gestione degli effetti delle carte personaggio si notano attributi booleani nominati ad esempio "add2StepsToMotherNature". Nell'ottica di una maggiore parametrizzazione del codice, essi potrebbero essere sotituiti con dei valori interi.

Dal documento in allegato viene detto che nella fase di preparazione vengono create 12 carte personaggio per poi estrarne 3 casualmente. Si suggerisce di avere un identificatore univoco che permetta di creare unicamente le carte personaggio strettamente necessarie. Inoltre, la modellazione delle stesse carte personaggio presenta gli stessi difetti citati in precedenza in merito agli attributi e metodi relativi al gioco avanzato. Nonostante i vari sottotipi di carte personaggio ereditino dalla medesima classe CharacterCard, esse non sono a loro volta sovratipi, ma insiemi di attributi di cui la maggior parte vengono ignorati. Si suggerisce di arricchire ulteriormente il modello per avere il minor numero di attributi non rilevanti possibili.

# Confronto tra le architetture

L’utilizzo di strutture dati coerenti con lo specifico comportamento che si vuole ottenere è un aspetto che dovremmo considerare maggiormente all’interno del nostro modello concettuale.

La nostra gestione dei turni potrebbe essere meglio distinta, e il controllo di fine partita venire fatto senza ausilio di variabili ma solo con un controllo a fine turno.

L'utilizzo delle classi Handler permette non solo migliore astrazione, ma permette alle classi di comunicare in modo più semplice evitando di esporre l'intero contenuto degli oggetti a cui fanno riferimento. Prenderemo in considerazione questa opzione per facilitare l'interazione fra classi.