Peer-Review 1: UML

Matteo Civitillo, Alessandro Paolo Gianni Callegari, Leonardo Chiaretti, James Enrico Busato

Gruppo AM19

Valutazione del diagramma UML delle classi del gruppo AM28.

# Lati positivi

UML molto chiaro e organizzato, metodi completi per la maggior parte e autoesplicativi.  
Positivo il fatto che si abbia già pensato nei dettagli il rapporto tra client e server e le divisioni dei vari compiti di ciascuno. Tra questi, molto interessante la gestione doppia del player lato server e lato client e come il lato server gestisca tutto ciò che è “pubblico” mentre il lato client gestisca solo i dati dell’area di gioco del singolo player. Questo permette di inserire anche un modo semplice per gestire un eventuale disconnessione del player, memorizzando tutti i dati in locale fino alla fine del turno.

# Lati negativi

L’utilizzo dei puntatori fa pensare più a un codice in linguaggio C che in Java e questo potrebbe portare ad ambiguità in seguito, consigliamo di sostituire i puntatori con oggetti indicati in UML.

Inoltre non è molto chiaro come vengono create le carte e non è chiaro neanche come avviene il calcolo dei punti: checkGoal() a che classi/metodi accede e come accede per calcolare i punti?

Non sarebbe meglio creare un oggetto counter per memorizzare le risorse nell’area di gioco di ciascun player al posto che dei singoli attributi gestiti dall’interfaccia?

Manca un attributo o un metodo che chiarisca la composizione di Personal Playground, sia per quanto riguarda PlacedCardList sia per quanto riguarda Card

# Confronto tra le architetture

La creazione di una classe astratta carta e di una classe astratta deck che vengono specializzate poi nelle rispettive sottoclassi era un’idea che avevamo avuto anche noi inizialmente ma che era stata poi accantonata perché ci siamo scontrati con un problema di implementazione delle carte. Come legate il fronte di una carta al suo retro? Come generate le carte?

Ci è piaciuta molto l’idea di creare un cliente non completamente “thin” ma che svolga tutte le operazioni del player e di lasciare al server il compito di coordinarle e di gestire i dati comuni.