

Peer-Review 1: UML

Filippo Raimondi, Luca Trifaro, Matteo Sala, Mattia Federico Volpi

Gruppo GC03

Valutazione del diagramma UML delle classi del gruppo GC50.

Lati positivi

Buon utilizzo del design pattern Strategy per la gestione degli algoritmi per il calcolo del punteggio degli obiettivi.

Generalmente le classi sono state suddivise bene, favorendo quindi un buon riutilizzo del codice.

Lati negativi

L'attributo "receiver" in Message potrebbe essere in più, in quanto ogni giocatore dovrebbe ricevere tutti i messaggi.

Utilizzare nomi più chiari e globali per i metodi e nomi variabili, per esempio non risultano autoesplicativi i metodi con i nomi CornersToCardsX e CornersToCardsY.

Si potrebbe "alleggerire" la classe Game creando un'ulteriore classe che rappresenti il tavolo di gioco.

Si potrebbero suddividere meglio le classi delle carte, differenziando meglio ogni tipo di carta.

La classe Chat è solamente un array, si potrebbe usare un ArrayList in Game.

Si potrebbe utilizzare il design pattern singleton se si volessero gestire più Game contemporaneamente.

Confronto tra le architetture

Nel nostro modello si potrebbe utilizzare il design pattern strategy per gestire in modo più compatto i differenti punteggi delle carte.

Nel nostro modello manca la funzione di PlayerStatus, che potrebbe invece servire.

Al posto di utilizzare un char come Game.status si potrebbe utilizzare un enum.