

# Peer - Review 1

## UML DIAGRAM

Valutazione del diagramma UML delle classi del gruppo AM01

### **Reviewer : AM44**

Camilla Bongiovanni

Riccardo Cadario

Tommaso Dri

Edoardo Alberto Fumagalli

## **Indice**

- 1. Analisi UML**
  - 1.1. Lati positivi
  - 1.2. Lati Negativi
  - 1.3. Confronto tra le architetture
- 2. Conclusioni**

# 1 Analisi UML

Di seguito esamineremo gli aspetti positivi e le criticità riscontrate tramite un'analisi approfondita della struttura compresa nell'UML in questione.

## 1.1 Lati positivi

È apprezzabile la completezza e l'attenta cura per ogni dettaglio. Ogni elemento, ogni classe, ogni relazione è stato modellato con precisione e chiarezza, fornendo una panoramica esaustiva e dettagliata del gioco.

La completezza di questo diagramma è rimarcabile: nessun aspetto significativo del sistema è stato trascurato; dalle classi principali alle relazioni complesse, tutto è stato accuratamente rappresentato.

Inoltre è da sottolineare l'attenzione ai dettagli: nulla è stato lasciato al caso; ogni elemento è stato modellato con la massima precisione.

È evidente che dietro questo diagramma c'è un lavoro diligente e una profonda comprensione del gioco e dei vostri obiettivi.

L'utilizzo dell'interfaccia CardFace potrebbe sembrare controintuitivo a prima vista: anche se per ora è implementata solo da una classe, può rendere più facile estendere il codice in futuro, implementando potenziali estensioni al gioco.

Avere più costruttori per la stessa classe con firme diverse, come per la classe FrontCardFace può essere utile per consentire la creazione di oggetti in modi diversi o per fornire opzioni di inizializzazione flessibili.

Notevole l'attenzione posta alle funzionalità avanzate, quali le partite multiple e la persistenza gestite dalla classe GameManager e la chat progettata ad hoc.

Infine, interessante il fatto che ciascun giocatore abbia la possibilità di scegliere il colore della propria pedina con la conseguente gestione dei casi in cui lo stesso colore sia conteso da più persone.

## 1.2 Lati negativi

- **Incapsulamento delle classi:** sebbene sia stata apprezzata la cura per i dettagli, d'altro canto in un UML iniziale è importante mantenere un livello di astrazione appropriato. Questo significa che l'UML dovrebbe fornire una visione generale e concettuale del sistema, piuttosto che un dettaglio eccessivo sulle singole classi o implementazioni specifiche.

Infatti abbiamo notato uno scarso utilizzo dell'incapsulamento delle classi e dell'astrazione: in molte classi si può evitare di esporre direttamente getter e setter, ma al contrario potevate implementare l'interfaccia delle classi in modo da fornire metodi che rappresentano operazioni significative o comportamenti dell'oggetto, come ad esempio nella classe Game.

Questi metodi dovrebbero incapsulare il comportamento interno dell'oggetto e fornire un'interfaccia più astratta e orientata al dominio del problema. L'obiettivo principale della progettazione orientata agli oggetti è quello di creare classi che nascondano i dettagli implementativi e forniscono un'interfaccia chiara e coesa per l'interazione con l'oggetto.

- **Classe MultiChoice:** l'utilizzo della classe MultiChoice appare superfluo: il problema della scelta del colore della pedina poteva essere gestito più facilmente, senza dover ricorrere all'implementazione di nuove classi specifiche. Si potrebbe semplicemente aggiungere un metodo all'interno di PlayerData che controlli la gestione dei colori delle pedine. Sarebbe più facile gestire la scelta del colore se il gioco proponesse solo i colori ancora disponibili. Per esempio dando la possibilità di scegliere in base all'ordine con cui i giocatori si sono uniti alla partita.

- **Classe BackCardFace:** non è stata ottimizzata al massimo la gestione della scelta del front/back delle carte da gioco, infatti sono state create due classi differenti che complicano l'implementazione.

- **Player:** a nostro parere si poteva non sdoppiare la classe Player, utilizzandone solo una e di conseguenza semplificando la scrittura del codice.

### 1.3 Confronto tra le architetture

Ottimizzazione e semplificazione sono state le parole chiave del nostro diagramma infatti la differenza principale si nota nell'abbondanza di classi, alcune delle quali sembrano essere eccessive e persino superflue. Questo può rendere il diagramma prolisso e complicato, offuscando la visione generale del sistema e complicando l'implementazione futura.

In particolare, l'abbondanza di classi non solo aggiunge complessità al diagramma, ma può anche suggerire un'eccessiva frammentazione del sistema. Molte di queste classi potrebbero essere amalgamate o rimosse del tutto senza compromettere la struttura o la funzionalità complessiva dell'UML.

Inoltre, l'uso di numerose strutture dati all'interno del diagramma, sebbene possa essere utile per modellare dettagli specifici, può portare a un'implementazione futura ingombrante e difficile da gestire. L'adozione di un approccio più snello e mirato potrebbe semplificare notevolmente.

In conclusione, pur riconoscendo il valore dell'approccio dettagliato nella modellazione UML, è importante bilanciare la completezza con la chiarezza e la concisione. Ridurre la quantità di classi e strutture dati superflue può contribuire a creare un diagramma più snello e comprensibile, facilitando così l'implementazione e la manutenzione del sistema nel lungo termine.

## 2 Conclusioni

Nel diagramma UML sono rappresentati tutti gli elementi che possono guidare i membri del progetto verso un'implementazione efficiente e funzionante. Il passo successivo, secondo il nostro parere, è domandarsi quali di questi elementi siano veramente necessari. Per ottenere una risposta, consigliamo di adottare lo stesso approccio suggerito dal professor Margara: ragionare sull'astrazione e considerare le interfacce. Per loro natura, sintetizzano in modo eccellente come una classe è percepita dall'esterno, da un utilizzatore; pertanto, rappresentano un ottimo strumento per comprendere con precisione e chiarezza il ruolo e i compiti di ogni oggetto, ovvero quali metodi deve implementare e quali no.

Ringraziamo i nostri colleghi per la puntualità nelle consegne e per i profittevoli spunti offerti dallo studio del loro diagramma.

Auguriamo un buon lavoro,  
AM44