

Peer-Review 1: UML

Gianmarco Schifone, Alessia Beatrice Viani, Leonardo Vicentini, Andrea Zhang

Gruppo 29

<https://github.com/leonardovicentini/ing-sw-2024-schifone-viani-vicentini-zhang.git>

Valutazione del diagramma UML delle classi del gruppo 19.

<https://github.com/federicovilla55/ing-sw-2024-sood-villa-vitali-zanzottera>

Lati positivi

1. **Suddivisione chiara in Package:** Il progetto presenta una suddivisione chiara in package, che favorisce la comprensione e la manutenibilità del codice. Questa organizzazione strutturale agevola la localizzazione delle componenti del sistema e facilita la collaborazione tra i membri del team di sviluppo.
2. **Uso adeguato dei pattern Strategy e State:** L'implementazione dei pattern Strategy e State dimostra un'attenzione alla progettazione del software basata su principi solidi e buone pratiche. Questi pattern consentono una gestione flessibile e modulare del comportamento degli oggetti, migliorando la manutenibilità e la scalabilità del sistema.
3. **Buona suddivisione del flusso di gioco nel proprio turno conforme alle regole del gioco:** Il sistema segue fedelmente le regole del gioco, garantendo che il flusso di gioco nel turno di ciascun giocatore avvenga nell'ordine corretto specificato dalle regole.
4. **Memorizzazione degli oggetti Message senza l'intero oggetto Player:** La decisione di memorizzare solo il nome dei mittenti e dei destinatari degli oggetti Message anziché l'intero oggetto Player contribuisce a mantenere leggera la struttura dei dati e a ridurre la complessità del sistema.
5. **Appropriato impiego di classe generica:** L'utilizzo della generic class per il Deck consente una gestione efficiente delle carte al suo interno.

Lati negativi

L'UML del gruppo 19 non presenta veri e propri lati negativi, ma si tratta solo di scelte progettuali prese. Di seguito vengono riportati alcuni **suggerimenti** in merito ad aspetti di possibile modifica o miglioramento:

1. Uno degli aspetti che richiede attenzione nell'attuale implementazione è la comprensibilità limitata nell'utilizzo dell'enum `Direction` e dei relativi metodi per il piazzamento delle carte. Potrebbe essere utile esaminare in che modo queste parti dell'architettura potrebbero essere rese più chiare e intuitive per facilitare la comprensione del design del software.
2. Dato che viene utilizzata solo una classe, si osserva che molte istanze presentano attributi o metodi che risultano superflui o non dovrebbero essere chiamati su una specifica carta, come ad esempio *`enoughResourceToBePlaced()`*. Questo è dovuto alla limitata varietà delle classi utilizzate.
3. Non esiste una distinzione logica tra la carta iniziale e la carta risorsa/oro, poiché entrambe le tipologie di carte sono trattate in modo simile.
4. Denominazione della classe poco intuitiva: nel documento è riportato: “la classe `Station` definisce il campo di gioco di un giocatore”, tuttavia riteniamo che il nome `Station` non suggerisca l'idea di campo da gioco rispetto a un nome più diretto come `PlayerArea`.

Confronto tra le architetture

Riportiamo di seguito alcuni delle differenze progettuali più significative:

1. La distinzione principale tra le due architetture risiede nell'approccio all'organizzazione delle carte. Nel nostro caso, abbiamo optato per una suddivisione delle classi rappresentative delle specifiche carte e l'utilizzo di classi astratte che riflettono il tipo di carta. Questo approccio mira a fornire una maggiore chiarezza e flessibilità nel codice, consentendo una gestione più granulare delle varie tipologie di carte. Tuttavia, tale decisione implica una rinuncia a vantaggi come la coerenza nel design del software e il risparmio di memoria, come invece garantito dall'architettura del gruppo 19.
2. Il gruppo 19 gestisce correttamente lo stato del gioco tramite una enumerazione (*SETUP*, *PLAYING*, *PAUSE* e *END*) e dei metodi opportuni in *Game*. Nel nostro caso abbiamo deciso di pensare in seguito alla gestione delle fasi di gioco, poiché abbiamo pensato al *Model* a partita già avviata.
3. Altro punto di forza del modello del gruppo 19 è il fatto che ciascuna carta ha una descrizione, che può risultare molto utile quando si andrà a progettare la *View*.