

# Peer Review 1: UML

Gruppo Revisionato 39: Piccoli, Ponzani, Piotti, Rodler  
Gruppo Revisore 49: Pucci, Ruggieri, Santarossa, Sciarretta

Aprile 2024

## 1 Aspetti Positivi

- **Gestione degli obiettivi:** Differenziare gli obiettivi in varie sottoclassi risulta essere un'utile scelta progettuale in quanto il calcolo dei punteggi ad essi associati segue modalità diverse. Corretta ci sembra essere anche la scelta di separare, tramite composizione, il concetto di obiettivo da quello di carta obiettivo. (Attenzione però che non tutti i tipi di obiettivo sono secondo noi necessari, vedi "Aspetti Critici")
- **Gestione delle card:** Condividiamo con voi la gestione della classe Card e delle sue sottoclassi. In particolare, ci sembra una buona scelta quella di separare le CardPlaying dalle CardObjective in modo da risultare evidente la differenza del loro comportamento 'statico'. Buono anche il fatto di considerare le CardGold come sottoclasse di CardResource e riferire poi entrambe alla sovra classe comune CardPlaying.
- **Sviluppo delle classi PlayingBoard e Player:** Le classi PlayingBoard e Player sono secondo noi ben sviluppate e complete in quanto incorporano in maniera esaustiva le regole e il flusso del gioco. Inoltre la classe PlayingBoard ci sembra strutturalmente adeguata a dialogare con il controller.

## 2 Aspetti Critici

- **La classe Corner:** La distinzione tra il corner di una carta risorsa e quello di una carta oro può essere un modo per semplificare l'implementazione, tuttavia tale scelta ci sembra ridondante in quanto questi possono essere gestiti in maniera analoga.
- **Composizione tra carte oro e obiettivi:** Non ci è chiara la presenza di una composizione tra le classi CardGold e Objective. A nostro parere, una carta oro non ha alcun legame strutturale con il concetto di obiettivo che anche una carta risorsa non abbia. Infatti, sebbene la classe CardGold abbia una sorta di "obiettivo" da realizzare per guadagnare punti, il punteggio che tale carta porterà (o meno) a

chi la gioca è da considerarsi un calcolo limitato al momento in cui la carta viene piazzata. Secondo noi è meglio tenere distinta tale logica dal calcolo dei punteggi delle CardObjective, il quale invece riguarda la configurazione finale di tutte le carte piazzate da ogni giocatore.

- **Pattern Significativi** Per una maggiore chiarezza espositiva, potrebbe essere utile evidenziare già nell'UML l'eventuale adozione di pattern significativi, quali lo strategy pattern, che sembra essere stato utilizzato per la gestione degli obiettivi tramite la classe Objective.

### 3 Confronto tra architetture

- La sostanziale differenza tra gli UML dei due gruppi riguarda la struttura dati scelta per implementare la classe PlayingStation di ogni giocatore. Avendo il calcolo degli obiettivi finali una forte connotazione geometrica, l'utilizzo di una Hashmap, da voi adottata, risulta sicuramente più elegante della nostra matrice, la quale però è di semplice utilizzo per gli algoritmi di posizionamento e calcolo dei punteggi.