

# Peer-Review 1: UML

Mattia Repetti, Davide Salonic, Jaskaran Ram, Lorenzo Reitani

Gruppo GC11

Valutazione del diagramma UML delle classi del gruppo GC01.

## Lati positivi

- Il diagramma è nel complesso ben idealizzato e organizzato. C'è una buona divisione dei ruoli delle varie classi implementate
- Buona l'idea di utilizzare dei tipi ad-hoc per gestire le tiles empty e forbidden.
- Buona anche l'idea di utilizzare un file JSON per gestire le carte obiettivo personale.
- Classi ricche di metodi sia per la gestione delle interazioni che per il controllo dello stato.
- Tutte le classi principali del model sono accessibili dalla classe Game.

## Lati negativi

- Manca una bozza di controller/view per far capire come gestire l'interazione
- Potrebbero essere aggiunte più eccezioni, ma andrebbe chiarito meglio chi le solleva
- Perché mettere i booleani forbidden ed empty in ItemTile se già i valori sono presenti nella enum ItemType?
- CommonGoalCard.achievedBy() potrebbe non prendere parametri e ritornare una lista di Player più che prendere un Player e ritornare un booleano
- Nelle common goal card non è chiaro se è presente un sistema per calcolare il punteggio da dare ai giocatori e come viene dato
- Tutti gli attributi sono public, sarebbe più sicuro metterli privati e creare dei metodi public get e set per evitare che possano sfuggire riferimenti e che gli attributi vengano cambiati da chi non deve.

## Confronto tra le architetture

- Utilizzare la classe turno per gestire meglio l'avanzamento dello stato del gioco è una buona idea.
- Si potrebbe implementare una classe Lobby per tenere traccia dei giocatori e del loro stato di connessione.
- Impostazione generale molto simile al netto di leggere il JSON per gli obiettivi personali. Utilizzo dello stesso design pattern per le CommonGoalCard