

# Peer-Review 1: UML

Mirko Calvi, Simone Calzolaro, Elena Caratti, Gabriele Clara di Gioacchino

Gruppo 5

Valutazione del diagramma UML delle classi del gruppo 14.

## Lati positivi

- Scalabilità: il fatto che l'UML sia molto modulare garantisce un'alta scalabilità in quanto permette facili modifiche e aggiornamenti senza alterare il codice.
- Lettura da file: facilita l'inizializzazione del gioco e predispone le basi per la persistenza.
- getInstance(): apprezzato l'utilizzo del pattern Singleton sul EndGameToken al fine di averne una sola istanza. Attenzione però per il multipartita

## Lati negativi

- Ereditarietà: usata in modo ambiguo. Le sovraclassi GoalCard e Token sono vuote quindi le loro sottoclassi non ereditano nulla. Inoltre le classi ObjectCard e le sottoclassi Token non hanno metodi in comune anche se ereditano dalla stessa classe (Token).
- ObjectCards: alcune classi riferite ad ObjectCard potrebbero essere compattate in un'unica classe.
- Implementazione di CommonGoalCard e GoalChecker: consigliamo una implementazione basata su una sola classe che implementa tutte le interfacce. Ogni interfaccia si occuperebbe di sviluppare dei metodi precisi per un determinato scopo. Ad esempio: un'interfaccia per il controllo della forma, una per la ricerca dei gruppi, una per il conteggio e una generica per le carte non classificabili nelle interfacce precedenti. Ponendo GoalChecker come tipo statico all'interno di CommonGoalCard invece non sarà possibile accedere ai metodi degli oggetti dinamici (ColorGoal, StructureGoal, PreciseStructure)

## Confronto tra le architetture

- ScoringToken: i nostri token sono di tipo enum poiché sono un numero finito. Questo garantisce una gestione dei token più semplice anche se minore scalabilità.
- Controller e Model: il nostro UML ha le classi Game e Player nel Controller e non nel Model. In questo modo il Model è totalmente passivo e sarà il Controller a chiamare i metodi per aggiornare il suo stato.
- Classe Model e Player: le funzioni di gestione del gioco di queste classi noi le abbiamo messe nel controller invece che nel model, come la gestione dei turni e delle azioni consentite alleggerendo il model.