

# Peer-Review 1: UML

Alessandro Pugnaghi, Gianluca Ruzzi, Christian Rossi, Kirolos Sharoubim

Gruppo GC21

Valutazione del diagramma UML delle classi del gruppo GC11.

## Lati positivi

Le classi presentate nell'UML hanno i seguenti lati positivi:

- Game: presenta tutti gli attributi e tutti i metodi necessari.
- Lobby: presenta tutti gli attributi e tutti i metodi necessari.
- Board: i metodi inseriti sono corretti (eccetto per getBag che non viene utilizzata).
- Bag: presenta tutti gli attributi e tutti i metodi necessari.
- Tile: la parte indicata è corretta se implementata come descritto nel pdf (con ID).
- Player: gli attributi e i metodi necessari sono presenti.
- Shelf: addTile è impostato correttamente come privato.
- Triplet: è una buona idea implementare i vari obiettivi personali come triplete.
- CommonGoalCard: l'idea di utilizzare uno strategy pattern è buona.

## Lati negativi

Le classi presentate nell'UML hanno i seguenti lati negativi:

- Game: manca la classe Turn a cui si riferisce. Manca un attributo che indica il giocatore con la sedia. Il costruttore dovrebbe ricevere una lista di giocatori (da lobby) e costruire i vari oggetti Player. Mancano setter e getter di endGame. Alcuni metodi restituiscono void invece che le variabili corrette (es.: getTurn).
- Lobby: sarebbe più corretto definire l'attributo "players" come una lista di stringhe, che viene poi passata al costruttore di Game. Quindi andrebbero modificati tutti i metodi per uniformarli a tale scelta. La direzione della freccia che unisce Lobby e Game dovrebbe essere invertita.
- Board: il riferimento a Bag sarebbe meglio spostarlo in Game piuttosto che direttamente dentro Board. Manca il costruttore (che crea una board vuota).
- Bag: manca il costruttore (che crea un sacchetto con tutte le Tile pescabili). Se si effettua la modifica alla Board bisogna collegare direttamente Bag a Game.
- TileColor: EMPTY potrebbe essere sostituito da null, che è predefinito nelle enumerazioni.
- Player: manca il costruttore. Le connessioni alle classi dovrebbero essere aggregazioni.
- Shelf: manca un metodo per controllare quando la libreria è piena (che può sfruttare freeSpaces) e freeSpaces dovrebbe essere pubblico. Manca il costruttore della libreria e la relazione di aggregazione di Tile.
- CommonGoalCard: bisogna sistemare i collegamenti tra le classi dello strategy pattern (es.: implements non extends). Tale classe deve essere un'interfaccia. Manca il costruttore.

## Confronto tra le architetture

La struttura generale del modello complica la gestione da parte del controller in quanto invece che esporre solamente l'interfaccia di Game vengono esposte anche interfacce di altre classi che dipendono da Game. Sarebbe quindi possibile migliorare il collegamento tra le classi per semplificare l'interazione del Controller con il Model.