

# Peer-Review 1: UML

Federica Di Filippo, Christian Di Diego, Carmine Faino  
Gruppo AM31

Valutazione del diagramma UML delle classi del gruppo AM04

## Lati positivi

Gli aspetti positivi dell'UML sono:

- Uso di molteplici pattern per alleggerire il codice;
- Buona gestione della modalità esperto;
- Hanno già impostato la gestione della comunicazione;
- Uso di "read only" per evitare che vengano modificati attributi involontariamente.

## Lati negativi

Abbiamo riscontrato i seguenti aspetti negativi, e abbiamo proposto alcune possibili soluzioni:

- Il codice non è distribuito in classi separate, ma accorpato in due grossi blocchi principali che svolgono tutte le funzioni. Alcuni metodi richiedono accesso ad attributi appartenenti a classi diverse; quindi, con un codice distribuito sarebbero rintracciabili più facilmente  
→ Una possibile soluzione sarebbe creare una classe per ogni gruppo di metodi che svolgono funzioni simili, ad esempio SchoolBoard potrebbe racchiudere tutti i metodi che aggiungono/tolgono studenti dall'Ingresso e dalla Sala
- Sono assenti alcune classi essenziali, quindi non sono presenti metodi fondamentali di gestione del gioco;  
→ Inserire ad esempio la classe Player semplificherebbe la gestione dei turni, inoltre renderebbe possibile impostare il nickname, il colore scelto per le torri ecc...
- Il metodo merge(Archipelago a1, Archipelago a2) è posto all'interno della stessa classe Archipelago, quindi, non capiamo come sia possibile modificare l'intera lista degli arcipelaghi presenti in GameState  
→ La possibile soluzione sarebbe spostare il metodo in GameState
- I nomi di molti attributi, classi e metodi non sono chiari e quindi rendono difficile la simulazione del gioco
- Non capiamo perché la classe Message che implementa Serializable contenga come attributo un array di byte  
→ Serializable riceve stringhe o oggetti che vengono trasferiti tramite stream, quindi non è necessario l'array di byte
- Sono presenti metodi "initialize" che potrebbero essere inseriti direttamente nel costruttore di GameState.

## Confronto tra le architetture

Un punto di forza di questa architettura è l'utilizzo del pattern Decorator per la gestione della modalità Esperto; quindi, valuteremo se adottare anche noi una strategia simile, visto che abbiamo riscontrato alcune difficoltà con l'implementazione di questa modalità.