

# Peer-Review 1: UML

<Alessandro Tischer>, <Mykhailo Shpakov>, <Giacomo Stefanizzi>

Gruppo <GC 46>

Valutazione del diagramma UML delle classi del gruppo <GC 56>.

## Lati positivi

La classe astratta StudentSpace rende le classi che ne derivano più uniformi tra loro e i metodi di aggiunta e rimozione degli studenti universali. Inoltre sfrutta il polimorfismo rendendo il gioco più flessibile per delle eventuali modifiche. Nel caso bisognasse aggiungere degli elementi nuovi che accettino studenti, lo si potrebbe fare con più agilità.

La classe Starter svolge un ruolo essenziale per la creazione del gioco in quanto definisce i parametri della partita. Inoltre permette di lanciare più partite in parallelo.

## Lati negativi

A nostro avviso, la lettura dello schema UML non risulta essere facile. La presenza dei commenti renderebbe più accessibile la lettura del diagramma. Inoltre mancano i nomi dei parametri dentro le firme dei metodi e molti degli attributi.

La classe Round potrebbe essere superflua in quanto ordinare i players è un'operazione che si può fare anche dentro Roundmaster.

Riteniamo che gli Studenti, che sono implementati come classe separata, non dovrebbero cambiare il tipo e diventare valori interi e soprattutto le istanze non dovrebbero essere eliminate ma spostate.

C'è confusione tra i tipi Student, Color, int nei metodi di spostamento degli studenti

Bag non viene considerata estensione di Student Space seppur abbia i metodi di aggiunta e rimozione degli studenti ed ha array di studenti come attributo.

Sono presenti dei metodi chiamabili dal client che richiedono il passaggio di parametri oggetto.

Non è chiaro se la mancanza del metodo "enableInfluence()" in ColorTracker sia voluta.

A nostro modo di vedere, avere l'informazione Player dentro ColorTracker potrebbe essere meno efficiente di avere l'informazione "TrackedColor" ( che rappresenterebbe il possesso del professore di quel determinato colore ) dentro Player.

L'informazione della posizione di Madre Natura è salvata sia nella classe Island che nella classe Mother Nature, il che risulta essere ridondante e potenzialmente causa di conflitti.

L'implementazione delle carte personaggio non è evidente. Per come è descritto dall'UML, non sembra possibile che le classi effetto possano ricevere i parametri di cui hanno bisogno ( e.g. isola su cui inserire uno studente ) a meno di un'ulteriore richiesta dal client, il che non è ammissibile.

## Confronto tra le architetture

L'insieme delle classi Starter, Creation, ServerApp andrà sicuramente implementato nel nostro progetto per completare il lato server.

La classe astratta StudentSpace sarà oggetto di discussione nel nostro gruppo per una possibile implementazione valutando pro e contro.