

Peer-Review 1: UML

Prova finale di ingegneria del software A.A. 2023-2024

Gruppo 22: Ruggiero Davide, Somaschini Chiara, Villa Francesco, Zani Beatrice

Valutazione del diagramma UML delle classi del gruppo 12

1 Lati positivi

- (i) L'utilizzo di una *GameLobby* per l'inserimento e la gestione dei giocatori, in modo da poter controllare la costruzione delle partite.
- (ii) Gestione della partita tramite *GameStates*.

2 Lati negativi

- (i) Il diagramma UML è risultato poco auto-esplicativo, certi aspetti non sono stati pienamente compresi o semplicemente interpretati nel modo corretto.
- (ii) Le strutture e le classi risultano eccessivamente complesse poiché poco diversificate. Ad esempio l'insieme dei metodi all'interno di *PatternCondition* potrebbe essere semplificato attraverso una diversificazione della struttura dei vari obiettivi.
- (iii) Non è risultata chiara dall'UML la metodologia per capire quali sono gli ultimi due turni giocati: quello da chiudere non appena un giocatore arriva a 20 punti e il turno finale.
- (iv) Gestione dei lati della carta come semplice chiave di una mappa, forse converrebbe usare direttamente una classe piuttosto che una mappa di mappe. Inoltre, ogni volta nella classe *Field* bisogna associare ad ogni carta il suo side in un'altra mappa.

3 Confronto tra le architetture

Le due architetture risultano molto diverse in quanto si è deciso di affrontare la costruzione della soluzione in modi diametralmente opposti. La struttura proposta dal gruppo 12 risulta costituita da macro classi contenenti metodi complessi per adempiere alle varie funzionalità di gioco; la nostra è invece composta da più classi ma con meno metodi al loro interno, in modo da focalizzarsi sulla costruzione di percorsi diversificati.

Consigliamo dunque, dal nostro punto di vista, di provare magari a ridurre la complessità andando a separare alcune funzionalità. Inoltre, secondo noi, la scelta di rappresentare il field come una hashmap attraverso le coordinate (x, y) dei vari angoli è particolarmente infelice, in quanto ogni angolo è definito come una mappa di una mappa di una coppia, costruzione che sicuramente complica la parte di programmazione.