

Peer-Review 1: UML

Enrico Tirri, Rebecca Maria Tacchi, Luca Prada, Giorgio Negro
Gruppo 36

29 marzo 2023

Valutazione del diagramma UML delle classi del gruppo 26.

1 Lati positivi

I GlobalGoal sono ben strutturati e permettono una buona estendibilità.

Le Tile sono ben modellate permettendo sia una estendibilità per il tipo (TileColor) sia per la rappresentazione della tile stessa (possibilità di avere Tile differenti ma dello stesso tipo).

GameBoard permette la realizzazione del "soggiorno" di arbitraria dimensione e forma.

Player permette una completa gestione delle informazioni riguardanti il singolo giocatore: un'ottima gestione dell'assegnamento dei "Token" derivanti dai GlobalGoal e una corretta e dinamica gestione del punteggio giocatore.

Game consente una ottima accessibilità di tutte le componenti di gioco, permette inoltre il corretto funzionamento della routine di gioco.

2 Lati negativi

TileSack mantiene un riferimento ad un array di interi seppur sia stato definito composizione di Tile. Vi è una incongruenza sugli oggetti gestiti da tale classe, essa gestisce infatti un array di interi ma è possibile ottenere da essa

oggetti di classe `Tile` tramite il metodo `pop()`.

`PrivateGoal` definisce un riferimento a coordinate (posizione riga-colonna) ma non al tipo di `Tile` che deve essere posizionata a tali coordinate.

`Shelf` è stata strutturata con una possibilità di espandibilità (è possibile assegnare una shelf di qualsiasi dimensione tramite costruttore), ma non è chiara la funzionalità dei metodi statici `getRows()` e `getColumn()`: essi dovrebbero ritornare la dimensione della shelf la quale però non essendo statica non è accessibile a livello statico.

In riferimento all'assegnamento della "Tessera di fine partita", si rileva una ridondanza di metodi e processi di assegnamento: corretta la possibilità tramite metodo apposito `first()` della classe `Player` di assegnare tale "Token", ridondante però la presenza del metodo statico `checkBoardGoal()` nella classe `GameBoard`, il quale introduce un livello di "logica" a livello di "modello" e la cui funzionalità si realizza già tramite il metodo `isFull()` della classe `Shelf`.

3 Confronto tra le architetture

In generale la struttura dei modelli presenta molti punti di contatto, vi è una simile gestione dei `CommonGoal`, `PersonalGoal` e `Tile`. Interessante la modalità di rappresentazione tramite mappa della board del soggiorno che ne permette una arbitraria realizzazione.