

Peer-Review 1: UML

Marco Alfano, Divine Boimah, Arianna Boisseau, Andrea Cerutti
Gruppo AM25

3 aprile 2023

Valutazione del diagramma UML delle classi del gruppo AM34.

1 Lati positivi

Abbiamo apprezzato l'ordine e la facilità di lettura del class diagram fornitoci, nonostante riteniamo ci siano alcuni dettagli omessi. La compattezza dell'UML presentatoci e la sua generale chiarezza, rendono la futura implementazione dei Listener molto agevole, così come l'interazione con lo strato di rete.

1.1 BoardSpace

Abbiamo apprezzato l'utilizzo della classe BoardSpace e dei relativi Optional sia perchè rendono più semplice la comprensione dell'UML, sia perchè migliorano l'implementazione del codice.

2 Lati negativi

2.1 CommonGoalCard

Il metodo `attribute(Library library)` utilizzato per assegnare il punteggio relativo al completamento di un obiettivo comune è invocato da `Player.updateScore()`, per una maggiore chiarezza è necessario aggiungere una relazione di associazione da Player verso CommonGoalCard.

2.2 Player

Manca il metodo `getNickname()` necessario a `GameController` per gestire la disconnessione di un giocatore. Non dovendoci essere un riferimento tra l'interfaccia di rete e la classe `Player`, `GameController` non ha modo di identificare i diversi giocatori.

2.3 Game

L'associazione tra `Game` e `GameController` dovrebbe essere invertita, con la possibilità di convertire l'associazione in composizione.

2.4 Board

Non risulta chiaro come venga utilizzato l'attributo privato `objectCards`, nel caso rappresentasse la bag, quindi venga inizializzato con le 132 `ObjectCard`, non risulta molto efficiente. Una soluzione migliore a quest'ultima potrebbe essere quella di creare le `ObjectCard` solo nel momento in cui vengono effettivamente inserite della board con il metodo `resetNeeded()` o con il costruttore.

2.5 Card

Riteniamo che l'utilizzo ripetuto del termine "Card" per riferirsi a classi completamente differenti tra loro riduca notevolmente la leggibilità dell'UML.

2.6 CommonGoalCard

Abbiamo notato che il diagramma è mancante dell'intera sezione dedicata alle singole carte obbiettivo comune. Per sopperire a questa mancanza basterebbe effettuare l'override del metodo `attribute` in modo da differenziale le diverse modalità di calcolo dei punti.

3 Confronto tra le architetture

In generale il cuore di entrambi i nostri approggi alla modellizzazione del gioco è il medesimo. Infatti le tre classi principali (`Game`, `Player` e `Board`)

presentato funzionalità sovrapponibili e si differenziano solamente per piccole scelte implementative.

3.1 Board

Riteniamo che un metodo del tipo `restoreLastTaken()` risulta molto utile nel separare la logica di controllo dalla rappresentazione nel model. Nella nostra architettura invece non è necessario poiché abbiamo scelto di compiere dei controlli di consistenza della mossa prima che questa venga effettuata.