

# Descrizione UML

Laura Puccioni, Viola Renne, Sara Resta

CG16

## 1 Introduzione

Abbiamo utilizzato il pattern Model-View-Controller con la visione del Model come "database" e del Controller come logica del gioco, il Controller decide quindi le transazioni sulla base degli input. Abbiamo dunque cercato di mantenere il più possibile questa distinzione, inserendo una buona parte della logica all'interno del Controller.

## 2 Model

### Introduzione

**Game** Game è una classe astratta che contiene tutte le informazioni relative alle altre classi del modello (infatti GameController farà sempre riferimento a Game). All'interno di Game troviamo la lista dei giocatori, il giocatore corrente, il giocatore che ha giocato per primo nel turno corrente, il riferimento a Table, il round, la CharacterCard attiva in questo momento, lo stato base (che contiene sempre BasicState), la lista delle AssistantCard, tre interi che vengono settati a seconda del numero di giocatori e GamePhase e TurnPhase che aiutano il GameController a gestire le azioni nelle varie fasi del gioco. Game è una classe astratta estesa da BasicGame, che non aggiunge funzionalità rispetto a Game, e da ExpertGame, che implementa l'utilizzo delle CharacterCard e dei coins.

**Player** Player è una classe astratta che contiene informazioni relative al giocatore. Al suo interno troviamo il nickname e lo wizard scelti dal giocatore all'inizio del gioco. Inoltre sono presenti l'insieme delle AssistantCard del giocatore, la AssistantCard giocata dal giocatore in questo turno e un riferimento alla SchoolBoard del giocatore. Player è esteso da BasicPlayer, che non aggiunge funzionalità, e da ExpertPlayer, che inserisce la gestione dei coins del giocatore.

**SchoolBoard** SchoolBoard contiene le informazioni relative alla plancia del giocatore (studenti nella entrance e nella diningRoom, professori e numero delle torri).

**Table** Table contiene le informazioni relative al tavolo da gioco e dunque troviamo un riferimento alla Bag, alle isole (GroupIsland), alla posizione di madre natura e alle CloudTile.

**Bag** Bag è una classe che contiene gli studenti rimanenti, ovvero quelli ancora non utilizzati nel gioco

**GroupIsland** GroupIsland è una classe astratta che contiene le informazioni relative ai gruppi di isole. Al suo interno troviamo un riferimento alle isole (SingleIsland) contenute in quell'agglomerato, un boolean che indica se madre natura è presente o meno, il giocatore influente, e la presenza di noEntryTile (funzionalità avanzata data da una delle CharacterCard). GroupIsland viene implementata da BasicGroupIsland, che non aggiunge funzionalità, e da AdvancedGroupIsland che aggiunge il conteggio delle noEntryTile e che gestisce dunque la presenza delle noEntryTile.

**SingleIsland** SingleIsland contiene solamente gli studenti presenti su quell'isola.

**CharacterCard** CharacterCard è una classe astratta che contiene alcuni metodi che vengono invocati da GameController al fine di gestire il flusso del gioco (per questo al suo interno troviamo un riferimento a Game) e le informazioni relative alla CharacterCard corrispondente (ovvero il costo della carta). CharacterCard è estesa da varie classi. In particolare, BasicState sarà lo stato che verrà mantenuto in BasicGame e non aggiunge funzionalità rispetto a CharacterCard (è come se indicasse che non è presente alcuna CharacterCard attiva in quel momento del gioco). Le altre classi, invece, aggiungono uno stato (se presente), dei metodi per modificare lo stato (se presente), fanno Override di effect (se hanno un effetto immediato), oppure modificano uno o più metodi che vanno ad alterare il flusso del gioco (come il calcolo dell'influenza, il controllo di quanto madre natura può muoversi, etc.)

### 3 Controller

**GameController** GameController è la classe principale del Controller che gestisce tutte le chiamate provenienti dai Client. Troviamo un riferimento a Game e due parametri che servono per impostare il gioco, ovvero la modalità scelta dal primo giocatore (base o esperta) e il numero di giocatori.