

Relazione del progetto

MDP A-L

JPokéBattle

<https://github.com/Viquess/JPokeBattle>



Realizzato da:

Bova Vincenzo 2106832

Feature implementate

Di seguito si elencano tutte le feature implementate all'interno del progetto, divise per categoria.

Feature comuni

Minime

- Implementati Bulbasaur, Charmander e Squirtle;
- Implementate le statistiche base di ogni Pokémon;
- Assegnate a tutti i Pokémon le due mosse di tipo neutro apprese al livello 1;
- Implementate le schermate Start, Battaglia, Cambio Pokémon, "You Win" e "Game Over";
- Adottato Java Swing.

Tipiche

- Implementate tutte le mosse dei Pokémon scelti, rispettando le loro meccaniche di funzionamento, dipendenti dai loro tipi, ma ignorando i cambiamenti di stato.

Extra

- Implementate animazioni, riproduzione di audio clip, ed altri effetti grafici (set multimedia).

Feature sviluppatore singolo

Minime

- Reso possibile affrontare una serie di NPC, fino alla prima sconfitta.

Tipiche

- Preservato lo stato dei Pokémon del giocatore nella serie di lotte.

Extra

- Implementate strategie per un comportamento "intelligente" degli NPC (set botness).
-

Sviluppo dell'applicazione

Per lo sviluppo dell'applicazione ho in primo luogo deciso di progettare il sistema di gestione dei Pokémon. Per realizzarlo ho utilizzato un sistema gerarchico di classi così strutturato:

- un'interfaccia **Pokemon** che definisce tutti i metodi necessari per istanziare un Pokémon;
- una classe astratta **PokemonImpl** che implementa l'interfaccia Pokemon e che contiene le implementazioni di tutti i metodi comuni tra i vari Pokémon;
- una classe che estende PokemonImpl per ogni tipo di Pokémon da implementare nel gioco.

All'avvio dell'applicazione i 12 tipi di Pokémon implementati vengono registrati in una HashMap che ha per chiave l'ID del Pokémon (ovvero il nome della classe) e come valore l'istanza della classe, in questo modo risulta semplice ottenere una lista di tutti i Pokémon esistenti ed ottenere per ognuno di essi una nuova istanza, mediante il metodo `newInstance()`.

Ciò, inoltre, rende molto semplice mettere mano al codice per modificare i Pokémon esistenti, in quanto sarà sufficiente sostituire le classi dei Pokémon attuali con quelle dei nuovi Pokémon ed implementare le risorse aggiuntive (sprite e suoni), senza dover metter mano al resto dell'applicazione.

Per la gestione dei tipi e delle mosse ho progettato un sistema basato principalmente su degli enumeratori, così strutturato:

- un enum **Type** che definisce tutti i tipi presenti nel gioco (NORMAL, FIRE, WATER...) e che per ognuno di essi stabilisce tre liste che indicano rispettivamente i tipi su cui quest'ultimo è superefficace, non molto efficace e non efficace;
- un enum **MoveCategory** che definisce i tre tipi di categoria di una mossa, ovvero PHYSICAL, SPECIAL e STATUS;
- un enum **MoveType** che definisce singolarmente ogni tipo di mossa presente nel gioco, specificandone il tipo, la categoria, la potenza, l'accuratezza ed il numero massimo di PP;
- una classe **Move** che rappresenta la singola mossa vera e propria e che ha come attributi il MoveType e il numero di PP attuali.

Anche in questo caso risulta estremamente semplice modificare o aggiungere nuove mosse, in quanto è sufficiente aggiornare la lunga lista presente nell'enum MoveType, senza dover mettere mano ad altre parti del codice.

* Per una questione di giocabilità, si è fatto in modo che le nuove istanze dei Pokémon non potessero avere determinate mosse (in particolar modo quelle di categoria STATUS) in quanto le feature richieste non ne prevedevano l'implementazione. Qualora si voglia rimuovere tale limitazione è sufficiente modificare la riga 122 della classe PokemonImpl, cambiandone il contenuto in `while (classe.hasMove(move.getMoveType()))`.

Per la realizzazione dell'interfaccia grafica ho scelto di utilizzare Java Swing, in quanto avevo già avuto modo di utilizzarlo in passato. Ho quindi strutturato l'interfaccia grafica a partire da una classe **Application** che estende JFrame e che può contenere tre JPanel differenti, ovvero tre menù:

- **StartMenu**, che rappresenta il menù principale dell'applicazione, con accesso al bottone per iniziare una nuova partita e alla classifica, che viene visualizzata mediante la classe **LeaderboardDialog**, che estende JDialog;
- **PokedexMenu**, che rappresenta il menù dove è possibile costruire la propria squadra, visualizzando le informazioni su ogni Pokémon;
- **FightMenu**, che rappresenta la schermata di lotta vera e propria tra i Pokémon.

Inoltre per semplificare il lavoro e la realizzazione di alcune animazioni ho deciso di realizzare tre nuovi component:

- **CustomButton**, che estende JButton e consente di istanziare più facilmente i tanti bottoni presenti;
- **AnimatedLabel**, che estende JLabel e contiene i metodi per alcune animazioni inerenti le label, in particolar modo per far apparire un testo in modo progressivo e per muovere la label in modo fluido;
- **AnimatedBar**, che estende JProgressBar e consente di creare una ProgressBar stondata, con il colore che cambia in base al valore di quest'ultima e con dei metodi utili per animarla, ad esempio il metodo per decrementarla in modo fluido;

Ho gestito il combattimento direttamente tramite l'interfaccia grafica stessa e tramite le classi **Team** e **Battle**. Inoltre è stato realizzato un metodo contenente il sistema d'intelligenza del NPC (FightMenu, riga 375), che stabilisce quale sia la miglior mossa per quest'ultimo, decidendo anche se sia il caso di cambiare Pokémon. Qualora l'NPC non dovesse avere più Pokémon liberi, apparirà dal lato mediante un'animazione, annunciando la vittoria da parte del giocatore e passando al prossimo NPC da affrontare. Qualora invece fosse il giocatore a non avere più Pokémon disponibili, l'NPC apparirà ugualmente dal lato annunciandone la sconfitta e, qualora il proprio punteggio merita di entrare in classifica, apparirà un InputDialog per inserire il proprio nome nella classifica, che ho deciso di salvare semplicemente in un file di testo.

Ho infine arricchito il progetto con font custom, colonne sonore (una per i menù iniziali ed una per la battaglia), effetti audio (suono al click dei pulsanti, versi dei Pokémon, suono della fuga) e varie animazioni (apparizione del Pokémon, apparizione degli NPC, label animate, progressBar animate...).

Risorse utilizzate e fonti

- Colonne sonore » [YouTube](#)
 - Effetti sonori » [Instant Button](#) e [YouTube](#)
 - Versi di ogni Pokémon » [Pokémon Go Fandom Wiki](#)
 - Sprites frontali e posteriori di ogni Pokémon » [Pokémon Database](#)
 - Sprite frontale dell'allenatore » [Pokémon Showdown](#)
 - Sprite posteriore dell'allenatore » [Bulbagarden](#)
 - Font Logo » [Text Studio](#)
 - Font » [FontStruct](#) e [FontSpace](#)
 - Backgrounds, bottoni e altre grafiche » Free Templates, [CraftBench AI](#), PhotoShop, tanta pazienza e la mia cara amica Melissa
-