



SAPIENZA  
UNIVERSITÀ DI ROMA

Università "Sapienza" di Roma

Facoltà di Informatica

**Corso:** Metodologie Di Programmazione

# Documentazione JPokeBattle

**Author:** Alessia Cassetta

**Matricola:** 2113909

Giugno 2024

# Indice

1	Feature Sviluppate	2
2	Decisioni di progettazione	3

# 1 Feature Sviluppate

- **Minimo**

- Implementati 55 pokemon di prima generazione, con le loro mosse base e statistiche.
- Assegnare a tutti i Pokémon le due mosse di tipo neutro apprese al livello 1.
- Implementare le schermate “start”, battaglia, cambio pokémon, “you win”, e “game over”.
- Adottata Java Swing
- Far affrontare al giocatore una serie di avversari NPC, fino alla sua prima sconfitta.

- **Tipico**

- Preservare lo stato dei pokémon del giocatore nella serie di lotte.
- Implementate tutte le mosse dei Pokémon scelti, rispettando le loro meccaniche di funzionamento dipendenti dai loro tipi, ma ignorando i cambiamenti di stato come avvelenamento, stordimento, etc.
- Implementare una schermata leaderboard che mantenga i 10 record migliori.

- **Extra**

- Set crescita:
  - \* Implementare punti individuali e punti allenamento che migliorino le capacità dei pokémon sulla base delle vittorie, aggregandoli appropriatamente.
  - \* Implementare i meccanismi di passaggio di livello ed evoluzione dei Pokémon, incluso l'apprendimento di nuove mosse a determinati livelli.
- Set battaglia:
  - \* Implementare strategie per un comportamento “intelligente” degli avversari NPC, per supportare un'esperienza di gioco appagante.

Queste Feature sono state implementate in modo incrementale, partendo dal Minimo e aggiungendo le funzionalità Tipiche e Extra. Ogni Feature è stata testata singolarmente, per garantire il corretto funzionamento del codice.

## 2 Decisioni di progettazione

Il progetto è basato su un'architettura a classi, in cui ogni classe rappresenta un'entità del gioco. Le classi principali sono:

- **Pokemon**: rappresenta un Pokémon, con le sue statistiche, mosse e punti esperienza.
- **Ability**: rappresenta una mossa, con il suo nome, tipo, potenza, precisione e punti mossa.
- **Coach**: rappresenta un allenatore, con il suo nome, i suoi Pokémon e il suo stato.
- **Battle Frame**: rappresenta una schermata, con i suoi elementi grafici e le sue azioni.
- **Battle**: rappresenta una battaglia, con i suoi stati e le sue azioni.
- **Change Pokemon**: rappresenta un cambio Pokémon, con i suoi stati e le sue azioni.

Ogni classe è stata progettata per essere il più possibile coesa e con un'alta coesione. Inoltre, è stata adottata l'ereditarietà per le classi **Pokemon** e **Ability**, in modo da poter creare facilmente nuovi Pokémon e nuove Mosse.

La schermata più importante è la schermata della **battaglia**, in cui il giocatore può scegliere la mossa da usare o cambiare Pokémon. Questa schermata viene implementata da **BattleFrame** che estende la classe **JFrame**, e contiene tutti gli elementi grafici e le azioni che si possono fare durante la battaglia. Essa implementa l'interfaccia **BattleEventListener** che contiene i metodi per gestire le azioni del giocatore durante la battaglia. **BattleEventListener** comunica con la classe **Battle**, che contiene la logica della battaglia e i suoi stati.