

# PROGETTO JPOKEBATTLE

**Corso: Metodologie di programmazione**

**Simone Comignani**

**Matricola: 2139861**

## Introduzione:

Prima di addentrarmi nello specifico di questo progetto, vorrei condividere alcune riflessioni personali su ciò che ho imparato durante la sua realizzazione. Io e il mio compagno di corso, Simone Descontus, abbiamo collaborato utilizzando [GitHub](#) per sincronizzare i nostri progressi. Ho apprezzato molto il modo in cui ci siamo divisi i compiti e come, grazie a una comunicazione efficace, siamo riusciti a integrare i vari pezzi del progetto senza intoppi. Questa esperienza mi ha insegnato l'importanza della collaborazione e della comunicazione nel lavoro di squadra.

## Features:

- **Features comuni:**

- **Minime:**

- Con l'ausilio di JavaFX abbiamo implementato varie schermate :
  - Menu principale
  - Nuovo salvataggio
  - Lista dei pokémon
  - Scelta degli sfidanti PvP
  - Scelta degli sfidanti PvE
  - Riorganizzazione dei pokémon
  - Battaglia
  - Apprendimento mosse
  - Evoluzione pokémon
- In questo progetto abbiamo voluto attenerci molto al gioco originale, quindi abbiamo inserito nei pokémon tutte le statistiche che servivano per poter fare i vari calcoli come per esempio chi colpisce per primo, quanta probabilità hai di mancare il bersaglio e così via... Ovviamente possono essere modificate in battaglia attraverso l'ausilio di varie mosse. Le statistiche in questione sono:
  - Punti salute (PS)
  - Attacco
  - Attacco speciale
  - Difesa
  - Difesa speciale
  - Velocità
  - Precisione
  - Elusione
- Implementazione di 34 pokémon in totale, ognuno di questi pokémon può evolversi (eccetto per l'ultimo stadio evolutivo).

### ➤ **Tipiche:**

- Implementazione delle mosse con reazione elementale rispetto al tipo per esempio la mossa braciore contro Charmander sarà poco efficace, mentre un pistolacqua sì.

### ➤ **Extra:**

- Set crescita:
  - Ogni pokémon è dotato di punti esperienza che crescono a seconda del pokémon che sconfiggono. Una volta che il pokémon avrà accumulato abbastanza punti esperienza avrà un level up che inciderà sulle sue statistiche.
  - Implementazione dell'apprendimento mosse, difatti quando un pokémon ha un level up potrebbe imparare una nuova mossa, se ha slot delle mosse liberi allora la impara senza problemi, se invece non ne ha allora apparirà una schermata dove l'allenatore dovrà scegliere quale mossa far dimenticare o se lo desidera non fargli imparare la mossa.
  - Implementazione di un sistema evolutivo pokémon. Ogni pokémon quando fa un level up se è di un livello abbastanza alto, chiederà all'allenatore a fine partita attraverso un interfaccia se vuole far evolvere il pokémon o meno.

## ● **Features due sviluppatori:**

### ➤ **Minimo:**

- Implementazione di un sistema di battaglia alla meglio di 3 (Solo PvP)
- L'intero progetto è stato fatto con l'IDE Eclipse e configurato per funzionare con i repository di [GitHub](#)

### ➤ **Tipiche:**

- Implementazione di una sezione di creazione degli allenatori
- Implementazione di un sistema di salvataggi degli allenatori basato su scrittura file con il formato JSON

### ➤ **Extra:**

- Implementazione del sistema di allenamento contro un NPC (Franco) con il quale potersi allenare, gli allenamenti non influenzeranno vittorie e sconfitte dell'allenatore

## Risorse Esterne:

- **JavaFX:**

Per la gestione dell'interfaccia grafica abbiamo deciso di utilizzare la libreria [JavaFX](#) gestendo gli fxml da un programma apposito chiamato [SceneBuilder](#).

- **Gson:**

Per la gestione dei salvataggi abbiamo utilizzato una libreria di Google chiamata [Gson](#) che permette la lettura e scrittura di file in un linguaggio strutturale JSON in modo semplice e veloce, in quanto in grado di convertire in oggetto il file prendendo come parametri il file e la classe che siamo interessati ad instanziare.

## Pattern Utilizzati:

- **Model-View-Controller:**

Questo pattern abbiamo deciso di utilizzarlo per gestire al meglio la libreria JavaFx, infatti nel nostro progetto all'interno della cartella *src* avremo 3 cartelle, la prima è il model dove all'interno possiamo trovare tutte le classi utili al funzionamento del programma in modo logico, nella seconda abbiamo view che è principalmente incentrata sulla parte grafica, per rendere il tutto più ordinato abbiamo pensato di dividere la view in 3 sottocartelle ovvero:

- *Css*: ci sono tutti i file di stile delle interfacce (.css)
- *Img*: ci sono tutti i file immagini che utilizzeranno i file di stile e controller
- *Fxml*: ci sono tutti i file di impaginamento dell'interfaccia (.fxml)

Ed infine abbiamo la cartella Controller dove al suo interno troveremo tutte le classi controller per ogni interfaccia.

- **Simple Factory:**

Abbiamo deciso di utilizzare questo pattern dichiarativo per poter gestire al meglio la creazione dei pokémon utilizzando una stringa senza dover chiamare un costruttore

## Costanti (Enum):

- **Tipo:**

Contiene tutti i tipi che le mosse o pokémon possono contenere. Ogni tipo ha dei tipi a cui sarà debole, resistente o immune, ed è proprio grazie a queste proprietà che si riesce a capire quando una mossa è superefficace contro un pokémon o no.

- **Statistica:**

All'interno di questo Enum possiamo trovare le statistiche classiche di ogni pokémon quali: velocità, attacco, attacco speciale, difesa, difesa speciale. Dopo di questa abbiamo delle statistiche che vengono utilizzate solo in battaglia per le mosse di stato ovvero precisione ed elusione.

- **Categoria:**

Contiene le possibili categorie di mosse quindi fisica, speciale o di stato.

- **Mossa:**

Contiene tutte le mosse che i pokémon possono usare durante il combattimento, e sono caratterizzate da un nome, la potenza di base, numero massimo di utilizzi, precisione e la categoria della mossa (Enum Categoria), quest'ultima è molto importante perché ci fa capire se una mossa è di stato, se così fosse andremo a focalizzarci su diversi attributi quali sono la statistica la quale avverrà la modifica di stato, il livello della statistica la quale avverrà la modifica di stato e un parametro nel caso la mossa abbia come bersaglio chi la usa.

## Classi:

### • **Pokemon:**

È una delle classi più utilizzate all'interno del progetto. All'interno della classe possiamo trovare molti attributi quali:

- Nome pokémon
- Tipi (dove puoi averne massimo 2 per pokémon)
- Mosse (ogni pokémon ne può memorizzare e utilizzare 4)
- Parco mosse (tutte le mosse che il pokémon può imparare)
- Livello del pokémon (più alto sarà più le sue statistiche saranno alte)
- Punti esperienza correnti
- Punti esperienza necessari per salire di livello
- Livello di evoluzione (indica il livello dove il pokémon proverà ad evolversi)
- Pokemon evoluto
- Stage di evoluzione (indica quanto un pokémon è evoluto)
- Prova a evolvere (indica se il pokémon vuole provare ad evolversi o meno. ogni aumento di livello sufficiente all'evoluzione il valore di questo attributo cambia)
- Punti salute in battaglia
- Punti salute massimi
- Velocità
- Potenza di attacco
- Potenza di attacco speciale
- Difesa
- Difesa speciale
- Precisione
- Elusione

Parlando invece per quello che può fare questa classe si può iniziare a introdurre i metodi principali:

- Attacco, che andrà a restituire la categoria della mossa utilizzata;
- hit che al suo interno avrà l'algoritmo del calcolo del danno;
- incassa che dovrà far subire il danno al pokémon;

Per poter gestire al meglio i livelli del pokémon abbiamo creato un'interfaccia chiamata crescita. Al suo interno si trova:

- gainExp dove a seconda di quale pokémon ha battuto otterrà dell'esperienza;
- LevelUp dove si andrà a gestire l'aumento di livello e l'aumento di statistiche con l'aiuto di altri due metodi:
  - CalcPs al suo interno troviamo l'algoritmo dell'incremento dei punti salute;
  - CalcStat al suo interno troviamo l'algoritmo dell'incremento delle statistiche.
- Evolve dove si andrà a gestire l'evoluzione del pokémon.

- **Pokemon specifici:**

In questo paragrafo non spiego il funzionamento di una sola classe, ma di molteplici, difatti, ogni pokémon presente nel gioco ha una sua classe, con il suo nome che estende “Pokemon”. All’interno di ognuno di essi si troverà un solo costruttore dove si inseriranno tutti i dati che formano quel pokémon.

- **BaseStat:**

Essendo una classe annidata all’interno di Pokemon, prende le sue statistiche e le rende alterabili durante una battaglia. Essa è caratterizzata dai seguenti attributi:

- Statistica (Statistica associata);
- MainValue è un valore statico di base;
- BattleValue è il valore della statistica a cui si fa riferimento durante battaglia.
- Modify Level è il livello di alterazione che può avere la statistica, il valore massimo e minimo, a cui può raggiungere è +6 e -6

Per modificare effettivamente la statistica, si utilizza un metodo chiamato modify che prende come parametro un numero intero X (compreso tra -6 e +6) e modificherà il battleValue in proporzione.

- **UsableMove:**

Questa classe ha lo stesso obiettivo di BaseStat ovvero rendere una mossa utilizzabile in battaglia. È caratterizzata dalla mossa associata, i ppMax che non sono altro che il numero di utilizzi massimi della mossa ed i pp, che sono quanti utilizzi di essa rimangono al pokémon.

### • Allenatore:

L'allenatore non è altro che il profilo di un utente. All'interno di esso si trovano:

- Nickname
- Squadra di pokémon (con un massimo 6 pokémon per allenatore)
- MainPokemon (Il pokémon che in quel momento è in prima linea)
- Numero di pokémon in squadra
- Vittorie
- Sconfitte
- WinRatio
- Ultimo Accesso

I metodi più utilizzati in questa classe sono:

- SetMainPokemon dove si attribuisce, attraverso un indice correlato alla squadra pokémon, la creatura da mettere in prima linea
- SetSquadra
- GetPokemonId, se gli viene dato come parametro il pokémon, restituirà un indice
- GetPokemonById esso compie l'inverso del metodo precedente.

### • Franco:

Franco è l'NPC con cui ci si può sfidare per allenare i pokémon. Questa è una classe che eredita "Allenatore". È dotato di diversi attributi:

- Strategia, lo aiuta a prendere diverse decisioni
- Enemy, contiene l'allenatore contro cui è Franco e lo aiuterà a generare la sua squadra tarata alla squadra dell'avversario
- MainEnemy contiene il pokémon in prima linea dell'avversario

Grazie a 2 attributi chiamati scambioConsentito () e statusCounter (massimo 6 mosse di stato ogni cambio pokémon) Franco eviterà ridondanze di mosse inutili.

Franco per poter decidere se cambiare o attaccare utilizza diversi metodi:

- Agisci. In questo metodo Franco decide che mossa fare: può essere una di stato se statusCounter lo consente, uno scambio se il parametro scambioConsentito lo permette oppure una per colpire.
- Cambia. Nel caso in cui Franco è in una situazione dove deve cambiare pokémon si chiamerà questo metodo e se vincerà il 30% metterà un pokémon superefficace contro il mainEnemy se così non fosse, ci sarà una scelta casuale.
- ControllaSquadra controlla se ha un pokémon che possa essere superefficace contro il mainEnemy



- **FactoryPkmn:**

È una classe che viene usata, come suggerisce dal nome, per l'implementazione del pattern simple factory. È composta da soli metodi statici, quali:

- "Crea", prende come parametro il nome o indice (nel pokedex standard) del pokemon e ne restituisce un'istanza nuova
- Random, genera pokemon casuali, questo metodo lo ritroviamo nella classe di Franco nella generazione della squadra
- AllPokemon, restituisce una lista contenente ogni pokemon disponibile nel gioco

- **SaveManager:**

Proprio come la classe FactoryPkmn è composta da soli metodi statici, che servono alla creazione, sovrascrittura e caricamento dei salvataggi, utilizzando la libreria

[Gson](#). Questi metodi sono:

- NewSave, il quale prende come parametro un allenatore, scorre l'array di tutti i salvataggi e mette il nuovo salvataggio in coda
- LoadSave prende come parametro un nickname, scorre tutti gli allenatori dell'array e se trova il nickname corrispondente lo restituisce
- Save, prende come parametro un allenatore e aggiorna l'array dei salvataggi sovrascrivendo l'allenatore corrispondente
- GetSave, restituisce tutti gli allenatori presenti nell'array dei salvataggi

## Interfacce grafiche:

- **Menu principale**

Questo è il menu principale, dove puoi scegliere di andare ad allenarti, combattere contro qualcuno oppure creare un nuovo salvataggio.



- **Nuovo salvataggio**

In questa schermata hai la possibilità di creare il tuo profilo formando la squadra che preferisci con 34 pokemon selezionabili e scegliere il tuo nickname  
N.B. Il nickname non può essere simile a quello di un altro player (non case sensitive) e la squadra deve essere di almeno un pokemon



- **Lista Pokemon**

Una volta cliccato il riquadro + nella schermata del nuovo salvataggio apparirà questa piccola finestra dove puoi scorrere un totale di 34 pokemon e selezionarne uno



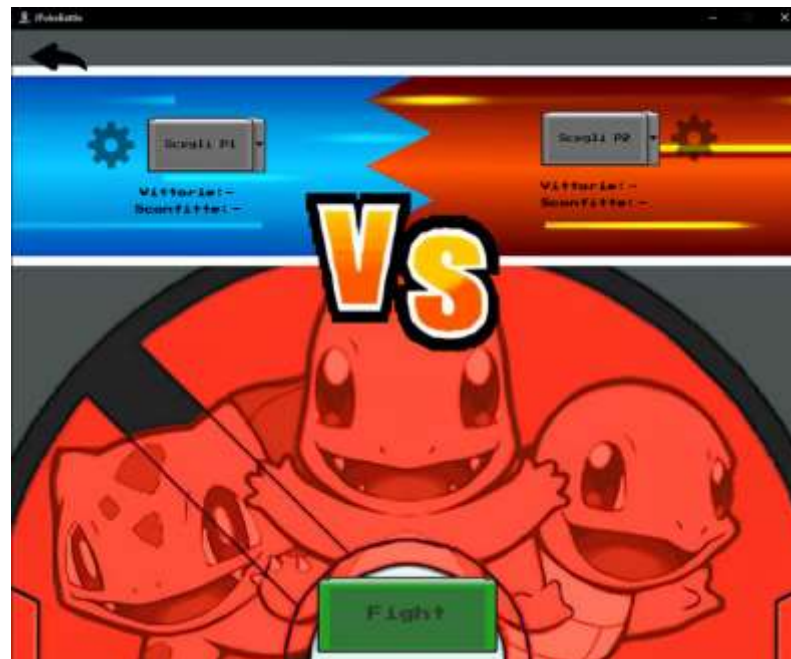
- **Allenamento con Franco**

Questa schermata è raggiungibile dalla schermata principale cliccando su allenamento, e puoi scegliere il tuo profilo con la possibilità di cambiare disposizione dei tuoi pokemon. Una volta cliccato su fight inizierà la battaglia



- **Scelta allenatore contro allenatore**

Questa schermata è accessibile dal menu principale cliccando su 1 contro 1, la schermata è quasi uguale a quella di Allenamento con franco, l'unico cambiamento è che si deve scegliere il secondo allenatore per combattere, anche lui può modificare l'ordine della squadra.



- **Riorganizzazione pokemon allenatore**

Questa è la schermata di riordinamento pokemon, è accessibile dalla schermata Scelta contro allenatore e da Allenamento con franco, quando si clicca sulla rotella delle opzioni vicino al nome dell allenatore, si aprirà questa finestra dove bisogna sostanzialmente cliccare i pokemon nella squadra nell'ordine che si vuole per riordinare la squadra. “Ma se mi sbaglio a cliccare devo chiudere e rifare tutto?” Per evitare di chiudere dopo un errore nel riordinamento, basterà cliccare nella squadra riordinata il pokemon che hai sbagliato ad ordinare, questo ritornerà nella tua squadra non ordinata.



- **Battaglia**

In questa interfaccia avviene il combattimento tra 2 allenatori o l'allenamento con Franco. Nel primo caso la battaglia una meglio di 3 e verrà gestita a scelte, prima sceglie il player 1 e poi il player 2. Nel secondo caso la battaglia è alla meglio di 1 il sistema di scelte cambia un po' rispetto al PvP dato che il player 2 è Franco, un "intelligenza" fatta da noi quindi dopo che sceglie il player 1 il calcolo del turno avviene subito.

Si può notare in foto come nel riquadro affianco al pokémon abbiamo un riquadro con all'interno il nickname dell'allenatore il numero di vittorie della meglio di tre, il nome del pokémon, il suo livello una barra della vita e dell'esperienza ed i suoi punti saluti correnti e totali, in più durante una partita PvP si ha il riquadro verde quando è il turno del player nel riquadro scegliere.

In questa interfaccia il player può scegliere tra 4 tipi di mosse (se il pokémon non ne ha imparate 4, le mosse che non ha vengono riempite con spazi bianchi), un cambio pokémon (se cliccata si aprirà l'interfaccia di cambio pokémon), una resa della partita e tutto il resoconto del turno viene scritto nel riquadro a destra bianco ogni turno.

Durante la battaglia ci sono diverse schermate che si possono aprire in modo automatico, quando il pokémon di un allenatore è esausto si apre la schermata di cambio pokémon, quando un pokémon vuole imparare una mossa ma ne deve dimenticare una si aprirà la schermata di apprendimento mosse e infine quando il pokémon si può evolvere si aprirà una schermata di evoluzione pokémon dove si potrà scegliere se farlo evolvere o no.





- **Scambia pokemon**

In questa schermata ci puoi entrare in 2 casi, o il player ha cliccato squadra e quindi vuole cambiare pokemon, oppure perché il suo pokemon è esausto ed è obbligato a metterne un altro in gioco. Nel primo caso la schermata è richiudibile perché non c'è un obbligo di scelta, nel secondo caso invece la schermata non è richiudibile, il giocatore deve scegliere un pokemon.



- **Apprendimento mosse**

Questa schermata viene aperta in automatico ogni fine turno se il pokemon dopo il lvlUp vuole imparare una nuova mossa e ne deve dimenticare una. Se si clicca una delle mosse che già conosce la dimentica e impara quella nuova, se si clicca la mossa che dovrebbe imparare, non la impara.

- **Evoluzione**

Questa schermata si apre in automatico ogni fine partita se il pokemon ha fatto un lvlUp ed ha un livello sufficiente per evolversi. Si visualizzerà una schermata del tuo pokemon e la sua evoluzione con 2 tasti, evolvi e non evolvere, se si clicca evolvi, il pokemon si evolverà se si clicca non evolvere invece non si evolverà ma al prossimo lvlUp richiederà l'evoluzione.

