

JPokeBattle

v0.0.3

Il progetto richiede di replicare il gameplay delle [lotte](#) all'interno dei videogiochi della serie Pokémon, facendo riferimento ai meccanismi definiti nei giochi della prima generazione (RBY).

[Video con un esempio di lotta.](#)

In breve [\[ref\]](#):

[...] una scena di battaglia a turni, in cui i Pokémon del giocatore e quelli avversari si affrontano. La schermata di gioco mostra le rispettive barre [PS](#) e un menu di opzioni, tramite il quale il giocatore può decidere di utilizzare uno dei quattro attacchi del Pokémon, [...], cambiare creatura [...]. Se entrambi i Pokémon attaccano, il primo a colpire è determinato dalla statistica velocità, sebbene alcune mosse [...] siano in grado di aggirare questa limitazione. Se uno dei due sfidanti sceglie invece un'altra opzione, questa verrà eseguita prima dell'attacco.

Ogni Pokémon utilizza [mosse](#) per ridurre a zero i PS dell'avversario, rendendolo esausto [...]. Se un Pokémon del giocatore ottiene la vittoria, riceve un numero di [punti \[allenamento\]](#) proporzionale alla forza del Pokémon battuto; dopo aver accumulato abbastanza esperienza un Pokémon sale di livello, incrementando le sue statistiche e apprendendo nuove mosse. Se il Pokémon del giocatore viene invece sconfitto, viene data la possibilità di mandare in campo un'altra creatura [...]. Se tutti i Pokémon della squadra sono esausti, il giocatore perde la lotta [...].

Valutazione

È possibile svolgere il progetto singolarmente o in un gruppo di due sviluppatori. Le feature richieste dalle due modalità di sviluppo, in aggiunta a quelle comuni, sono specificate di seguito. È possibile ottenere il punteggio massimo sia come singoli che in gruppo. Le feature richieste sono divise in tre fasce, che corrispondono a tre range di voto: minimo (0–18), tipico (19–26), ed extra (27–30L). Per esempio, per ottenere un voto nel range 27–30L è necessario implementare almeno parte delle feature definite nella fascia extra, in aggiunta a tutte quelle delle fasce precedenti. Il voto finale, all'interno di ciascun range di voto, verrà deciso sulla base della qualità dell'implementazione, della documentazione del codice, del diagramma delle classi, e della relazione.

Codice di condotta

Mi aspetto che ciò che consegnate sia frutto del vostro lavoro individuale. È difficile dare linee guida specifiche sul plagio, ma come “rule of thumb”: se non è stato digitato da voi, non è frutto del vostro lavoro. È concesso scambiare idee, ma non condividere e commentare codice. È concesso avvalersi di risorse e strumenti che vi aiutino nello sviluppo del codice e ricercare soluzioni online. È scoraggiato ma non impedito l'uso di librerie esterne a quelle fornite dalla

libreria standard: la valutazione verterà sul codice e le funzionalità sviluppate da voi. Consegne considerate plagio o in altro modo non frutto del lavoro dello studente verranno registrate come esami falliti e segnalate al comitato etico di facoltà.

Specifiche

Feature comuni

Minimo

- Implementare Bulbasaur, Charmander, Squirtle
- Implementare [statistiche base](#) dei pokémon
- Assegnare a tutti i Pokémon le due mosse di tipo neutro apprese al livello 1 (growl e tackle per Bulbasaur, growl e scratch per Charmander, tail whip e tackle per Squirtle)
- Implementare le schermate “start”, battaglia, cambio pokémon, “you win”, e “game over”
- Adottare Java Swing o JavaFX per la GUI

Tipico

- Implementare tutte le mosse dei Pokémon scelti (si legga la sezione “risorse”), rispettando le loro [meccaniche](#) di funzionamento dipendenti dai loro [tipi](#), ma ignorando i cambiamenti di [stato](#) come avvelenamento, stordimento, etc.

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 (rispettando il vincolo delle 4 mosse a disposizione di ciascun pokémon nella lotta: l'allenatore deve decidere, nel momento in cui una nuova mossa diventa disponibile ed il pokémon già ne conosce 4, se apprendere la nuova dimenticandone una delle precedenti a scelta)
- oppure, set multimedia:
 - Implementare animazioni, riproduzione di audio clip, ed altri effetti grafici
- oppure, set programmatore:
 - Implementare unit test del backend
 - Creare un Jar eseguibile

Feature - sviluppatore singolo

Single player mode

Minimo

- 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 (e.g., se un Pokémon riceve 10 PS di danno nella prima lotta, partirà con 10 PS in meno nella seguente, etc.)
- Implementare una schermata leaderboard che mantenga i 10 record migliori (numero di lotte vinte in serie)

Extra

- Set botness:
 - Implementare strategie per un comportamento “intelligente” degli avversari NPC, per supportare un’esperienza di gioco appagante
- Oppure, set TUI:
 - implementare una modalità di gioco testuale via terminale, selezionabile in alternativa alla GUI con un’opzione al lancio dell’applicazione

Feature - due sviluppatori

Two-player mode

Minimo

- Permettere la lotta tra due giocatori umani, al meglio delle 3 lotte; lo stato dei Pokémon deve venire ripristinato alla partita successiva (e.g., se un Pokémon riceve 10 PS di danno nella prima lotta, partirà con 10 PS in meno nella seguente, etc.)
- Sviluppare usando un repository git privato su github, bitbucket, o altra piattaforma analoga

Tipico

- Implementare la gestione di profili utente, nickname, conteggio delle partite giocate, vinte, e perse
- Implementare il salvataggio su file dei profili utente, ed il loro caricamento nella schermata “start”

Extra

- Implementare il set crescita di feature comuni; aggiungere una modalità di gioco nella quale i giocatori possono selezionare fino a 6 Pokémon, da allenare in lotte future, salvabili e ricaricabili assieme al profilo utente

Risorse

- Potete fare riferimento a [questo link](#) le mosse di ciascun Pokémon, guardando le mosse della generazione 1 apprese tramite level-up
- Potete utilizzare [sprite](#) ed [altre](#) risorse simili come avatar dei Pokémon; d'altra parte, non è necessario utilizzare immagini di Pokémon, ma anche di altre entità come animali etc. e relative immagini di dominio pubblico
- Lo stesso vale per clip [audio](#)
- Il marchio pokémon, il funzionamento dei videogiochi della serie, e le risorse associate sono proprietari delle rispettive compagnie. Pertanto, è vietata la loro distribuzione. Le consegne avverranno privatamente.

Consegna

Da consegnare, tramite google classroom:

- Il progetto eclipse del gioco in formato ZIP, con tutte le cartelle relative a codice sorgente e risorse necessarie all'esecuzione; la classe "JPokeBattle" deve contenere il main del gioco; il progetto deve contenere una cartella "docs" contenente la documentazione completa generata con javadoc
- il diagramma delle classi in formato PDF. Potete utilizzare strumenti di disegno come draw.io, Google Slides, PowerPoint, etc.
- Una relazione (individuale anche se si è scelto di sviluppare il progetto in gruppo) in formato PDF, che:
 - includa almeno nome, cognome, e numero di matricola,
 - Dichiarare quali feature sono state implementate
 - discuta
 - le decisioni di progettazione
 - i design pattern adottati
 - le risorse esterne utilizzate e la loro integrazione (se presenti)
 - l'applicazione di paradigmi di programmazione generica, funzionale, e stream.

Si faccia riferimento [al post di classroom](#) per le tempistiche di consegna e la contestualizzazione del progetto nella valutazione dell'intero esame.

Note