

Durante las últimas tres semanas, hemos estado desarrollando un programa, que simula un juego de la franquicia Pokémon. Este programa necesita un archivo .txt, donde venga información del combate que se quiere realizar. Dicha información requiere de dos entrenadores que cuentan con un equipo de Pokemons. Al pasar los datos de estos Pokemons, deben de ir acompañados sus estadísticas tal como se muestra en el siguiente ejemplo:

```
{Nombre entrenador}
```

```
{Nombre_Pokemon}{Type: {tipo}, Level: {nivel}, Atk: {daño}, Def: {defensa}, HP: {vida}, Agi: {velocidad}, {Atributo especial de cada tipo}: {valor}}
```

```
{Nombre_Pokemon} ( ... )
```

...

Exceptuando el atributo especial, los valores numéricos que se deberán pasar deberán ser números enteros positivos (int). El atributo especial varía según el tipo:

Planta -> Blessing (float) Fuego -> Temperature (float) Agua -> surge mode (bool : {False})

Este programa se ejecutará en una terminal.

Los cuatro archivos (los tres de código (main.py, pokemon.py y trainer.py) y el .txt) deben estar en la misma carpeta contenedora. Debes ir moviéndote en la terminal con los siguientes comandos hasta llegar a esa carpeta:

ls para ver en que carpeta y que archivos y carpetas hay.

cd Nombre_carpeta para moverte a otra carpeta hasta llegar a donde tengas los archivos.

Una vez que estes en la carpeta contenedora, al estar escrito en Python, tendrá que ejecutarse con Python3, aportando el archivo main.py, y el archivo .txt que contiene los datos de la batalla.

Un ejemplo para ejecutarlo con el archivo battle0.txt es:

```
- python3 main.py battle0.txt
```

Respecto al código, para empezar con el desarrollo de este, empezamos creando el archivo pokemon.py.

Primero de todo creamos la clase pokemon, definiendo sus atributos y métodos, como is_debilitated, y tras esto creamos las sub-clases FirePokemon, WaterPokemon y GrassPokemon, cada sus métodos y atributos, como puede ser para la sub-clase WaterPokemon, watter_attack.

Una vez creado todo esto, y brevemente documentado, pasamos a crear, en un nuevo archivo (Trainer.py), la clase Trainer, en la que creamos al entrenador, y le damos atributos y métodos, como puede ser select_next_pokemon o all_debilitated.

Una vez hechas todas las clases y sub-clases, fue hora de ponerse con la batalla (main.py).

Primero empezamos modificando el código para que se creen los pokemons.

Después nos pusimos a crear funciones que son necesarias para el funcionamiento de nuestro código como puede ser en nuestro caso la función ronda o inicio batalla.

Tras esto escribimos la secuencia de batalla teniendo en cuenta si el otro Pokemon se debilitaba o no, y comprobando cuando se debilitaba todo un equipo Pokemon para terminar la batalla.

Mientras tanto, hicimos que los datos como el daño y la curación se fueran guardando respecto a cada Pokemon, para una vez terminado el combate, crear un DataFrame con los datos guardados.

A continuación, hicimos un análisis con los datos del DataFrame, haciendo que se vayan guardando en otros DataFrames e imprimiéndolos por pantalla.

Tras esto fuimos leyendo todo lo que devolvía por ronda y asegurándonos que iba bien y si no iba bien corrigiéndolo. Lo mismo cuando saltaban errores de código y demás.

Para terminar, documentamos con docstrings todas las funciones, clases, métodos...

#Si se ejecuta el archivo unit_test_pokemon.py:

Algo a resaltar es que la función `check_surge_mode_activation(self)`, usa el método `check_surge_activation()`. Este método no está de la misma forma definida en nuestro código. Nosotros la definimos como `check_surge_mode()`, por tanto, es importante cambiar esto en las líneas 276 & 281.