Practica 1

Pablo Chantada Saborido(pablo.chantada@udc.es)

Pablo Verdes Sánchez(p.verdess@udc.es)

Manual de Usuario

DEPENDENCIAS

Para ejecutar el código fuente es necesario tener instalado la librería **Pandas** y una versión de **Python 3.10** o superior, para instalar la librería se puede utilizar en el siguiente comando en la consola:

pip install pandas

EJECUCIÓN

1. Descomprimir el archivo zip
2. Ejecutar el archivo main.py, para ello abrimos una consola de comandos y escribimos: python main.py archivo_a_utilizar.txt

Para esta práctica, con los archivos proporcionados podemos ejecutar:

python main.py battle0.txt *(archive corto)*

python main.py battle1.txt *(archivo largo)*

RESULTADOS

Por en pantalla (consola de comandos) se mostrará las peleas entre los Pokémons y al final las estadísticas:

1. Individuales
2. Por tipo
3. Tipo vs Tipo

Memoria

Para la creación de las estadísticas creamos una clase llamada BattleStats, que permite: almacenar los datos de las batallas y calcular la media y desviación estándar de las mismas. Además, el archivo main.py está dividido en varias funciones para una implementación más sencilla, manejable y sostenible:

- special_attacks (attacker, defender): realiza un ataque especial según el tipo de pokemon; wáter_attack, fire_attack o grass_attack
- normal_attacks (attacker, defender): realiza un ataque básico común a todos los pokemons
- determine_attacker_defender (p1, p2): selecciona que Pokémons es el atacante según su agilidad
- perform_round (attacker, defender, round_counter): realiza una ronda con el atacante pegando primero, en rondas impares se usan ataques especiales
- fight (p1, p2): realiza una pelea entre dos Pokémons mientras ambos estén vivos, los resultados se muestran por pantalla

- main (): lee el archivo de texto y realiza la batalla entre los entrenadores; durante el proceso se muestra por pantalla (consola) la evolución de la batalla. Al final, cuando a uno de los entrenadores no le quedan Pokémons, se termina el programa

1º Semana

Análisis del código y creación de “esqueletos” de clases como Pokémon, Trainer, etc. Además, conseguimos superar todos los test del archivo unit_test_pokemon.py

2º Semana

Pulir las clases creadas y generar las batallas correctamente en el archivo main.py

3º Semana

Creación de las estadísticas y documentación del código, tanto mejorar comentarios en línea como generación de Docstrings

4º Semana

Creación de la memoria y arreglos en el cálculo de las estadísticas, exactamente en el cálculo de la curación ya que interpretamos que, si el Pokémon no perdió vida, es decir, su vida actual es igual a su vida máxima; al realizar la curación esta “no se realiza” (cura 0 puntos de vida al Pokémon).