

Pokémon Battle Simulator:

This Python program simulates Pokémon battles and tournaments using a dataset of Pokémon stats. The simulation includes features such as stat scaling by level, type effectiveness, and turn-based battle mechanics.

Features:

The program includes Pokémon stats scaling to adjust stats according to their level, type effectiveness to calculate damage multipliers based on Pokémon types, a battle system to simulate battles between two Pokémon considering type effectiveness and stat-based mechanics, tournament mode to automatically conduct a tournament with four randomly selected Pokémon to determine a champion, and random Pokémon selection to choose Pokémon from a dataset and scale their stats based on random levels.

Requirements:

The program requires Python 3.7 or higher and the pandas library, which can be installed by running the command `pip install pandas`.

Dataset:

The program uses a `pokemon.csv` dataset containing Pokémon stats. Ensure the file includes the following columns:

- name: Name of the Pokémon
- type1: Primary type of the Pokémon (e.g., Fire, Water)
- hp: Base hit points
- attack: Base attack stat
- defense: Base defense stat
- speed: Base speed stat

Make sure the `pokemon.csv` file is in the same directory as the script.

Usage:

- Load the dataset: The script reads Pokémon stats from `pokemon.csv`.
- Battle Simulation: Use the function `battle(pokemon1, pokemon2)` to simulate a battle between two Pokémon.
- Tournament: Call the `tournament()` function to conduct a tournament with four randomly selected Pokémon.

Example Output:

The program provides detailed outputs, including Pokémon stats before the battle, damage dealt in each turn, type effectiveness multiplier, and the winner of the battle or tournament, with the `scale_stats(base_stats, level)` function scaling Pokémon stats based on their level.

The program includes the `get_type_effectiveness(attacker_type, defender_type)` function, which returns the damage multiplier based on type effectiveness, the `Pokemon` class representing a Pokémon with attributes like name, type, level, and stats (hp, attack, defense, speed), and methods such as `take_damage` and `is_fainted`; the `select_random_pokemon()` function, which randomly selects a Pokémon from the dataset and scales its stats; the `battle(pokemon1, pokemon2)` function, which simulates a battle between two Pokémon to determine the winner; and the `tournament()` function, which conducts a four-player tournament to determine the champion.

To run the program, execute the script directly using the command `python pokemon_simulator.py`, which will load the dataset and start a Pokémon tournament.

Notes:

Modify the `effectiveness_chart` in `get_type_effectiveness` to include additional type matchups.

Customize level ranges in `select_random_pokemon` by adjusting the random level generation logic.

Example Dataset (Sample)

name	type1	hp	attack	defense	speed
Bulbasaur	Grass	45	49	49	45
Charmander	Fire	39	52	43	65
Squirtle	Water	44	48	65	43

What Happens When You Run the Program:

When the program is run, it will automatically start a **Pokémon Tournament** with 4 random Pokémon. The program selects 4 random Pokémon from the dataset, assigns them a random level between 30 and 70, and scales their stats accordingly.

- **Battle Simulation:**
 - The Pokémon battle occurs between two randomly selected Pokémon at the beginning of each round.
 - The **Type Effectiveness** (e.g., Fire vs. Grass) is calculated based on predefined effectiveness rules, which are simplified (e.g., Fire beats Grass, Grass beats Water).
 - **Turn Order** is determined by comparing their **Speed** stats; the Pokémon with the higher Speed attacks first.
 - The Pokémon's **HP** is updated after each attack, and the battle continues until one Pokémon faints (i.e., $HP \leq 0$).
- **Battle Output:** The program prints the status of the Pokémon and the battle as it progresses:
 - Each Pokémon's current stats (HP, Attack, Defense, Speed).
 - Damage dealt by each Pokémon during their turn.
 - Notification of when a Pokémon faints.
 - The winner of each individual battle.
 - The champion of the tournament after all rounds

How to Interpret the Output:

The output will show a detailed description of each round of the battle, like this:

- **Battle Start:** The two Pokémon involved in battle are introduced with their stats.
- **Type Effectiveness Multiplier:** Displays how much more effective one Pokémon's attack is due to their type advantage.
- **Damage Info:** Each Pokémon's attack damage is calculated and displayed. You'll see how much damage each Pokémon does and their remaining HP.
- **Winner Announcement:** When one Pokémon's HP reaches 0, the other Pokémon is declared the winner of that battle.

After the individual battles, the final winner (the champion of the tournament) will be announced.

Program Flow

- **Tournament:** The program conducts a 4-player (2 rounds) tournament.
 1. Two battles take place between four randomly selected Pokémon.
 2. The winners of those two battles then face off in the Finals to determine the tournament champion.

Method/Function	Primary Author	Techniques Demonstrated
scale_stats	Samuel adu	Scaling formulas with optional arguments
get_type_effectiveness	Carlton Carter	Dictionary-based logic for type matching
Methods in Pokemon class- Take damage, is fainted Select random pokemon	Samuel Adu	Use of a class in a flexible manner
select_random_pokemon	Carlton Carter	Randomized selection and scaling logic
Battle	Samuel adu	Sequence unpacking
main	Samuel adu	ArgumentParser class
__str__ (pokemon)	Carlton Carter	F-strings counting expressions
simulate_launch	Carlton Carter	List Comprehensions

Vadivelu, N. (2018, June 24). *Complete competitive pokemon dataset*. Kaggle.
<https://www.kaggle.com/datasets/n2cholas/competitive-pokemon-dataset/data>