Introduction to Python & Programming

Hacker Night 2

Gaël Guibon

Essec - BBA - B2

20/11/2019 16:30 - 22:45

Subject

Browse and Query Pokémons

Objectives

- Create a program that will handle Pokémons data

1. Implement **required actions**
2. Merge them into a terminal interface (input)
3. Be creative and create **creative actions**

Required Actions: todo list

Todo List:

1. get the total number of pokémons
2. get the total numbers of pokémons for each generation
3. get the name of a pokémon based on its pokedex number (**Number**)
4. get a pokémon's values based on its name
5. get the number of legendary pokémons and their names
6. get pokémons names by primary type (**Type_1**)
    - example: Grass pokémons

7. view all the body styles and the number of pokémons per body style
8. get the name of the biggest pokémons
9. get the names of the rarest pokémons
10. get the most common pokémon for each type
11. get the heaviest pokémons and the fastest ones among them
12. get the most powerful pokémon by primary type (**Type_1**)
13. get the slimest pokémons by using its height and weight

14. get the strongest and the weakest pokémons that can mega evolve
15. get the thoughest pokémons and the ones that deal the biggest damages
16. get the most common type combinations
17. get the color frequencies for each type combination
18. get the global probability for pokémons to be male of female (**Pr_Male**)
19. simulate a (1vs1) "fight" between arbitrary pokémons
    - turn based
    - use pokémon stats as damage factors

Required Actions: textual interface and export

- Each todolist item should be able to print result or export them into a file
- Each todolist item should be triggered through a terminal query interface

Creative Actions

Creative actions example ideas:

- Create random pokémons by mixing existing data
- Create your own pokémon from scratch !
- Make your pokémon fight against a random one

```
import random
random.shuffle(myList) # shuffle a list inplace
```

- Add custom metadata and queries
  - example: be able to easily find your custom pokémons
- Stats queries : show all kinds of information from basic stats (counts, average, etc.)

Format

- **Individual**: no chatting
- Internet allowed (no chatting between groups!)
- **Scheduled breaks** (snack, nap)
  - 17h45-18h15
  - 19h30-20h
  - 21h15-21h45
- Otherwise **stay in class**

Evaluation

- Diversity of notions used
- Todolist completion
- Creativity actions
- Code clarity and good practices

Project delivery by email before leaving

Advises

- Use local **python install** + **visual studio code**
- Use `pprint.pprint()` to better print a dictionary

```python
import pprint
myDict = {'greeting': 'hello B2'}
pprint.pprint(myDict)
```

- A problem with your machine set up? Ask for help!
    - ...but for the code you are on your own.

Good luck!