Project 2 – Pokemon Lab

This is a team project with a “choose your approach” flavor. The program will access the pokemon API (https://pokeapi.co/) and produce data outputs depending on your choices.

Specifications:

- Teams of two or three are to be before next class.
- At least **100 points** needs to be gathered. It’s the same number for a team of 2 or 3.
- There are dependencies betweens step, each dependencies need to be completed before going to the next one.
- During the last class, the teams will come show the project to the teacher. The program needs to be executed in front of the teacher.
- Respect the outputs shown in the examples. It needs to be **identical.**

Considerations:

Contrary to the first project, it is expected that students will do their own research to find their own solutions. The teams will need to come up with a plan and apply that plan to succeed.

The chosen paths should all be executed from the “main.py” file in one execution. For example, if I choose 2a, 3a, 4a and then 2c. I would see the result as follow being printed:
-> I choose Jigglypuff ... (1)
-> There is currently X numbers of pokemon in the world (2a)
-> Y pokemons weight more than Jigglypuff (3a)
-> The first ability of Jigglypuff is X and Y pokemons has the same abilities (4a)
-> Jigglypuff fights for its health against [Random Pokemon] and wins! (2c)
...

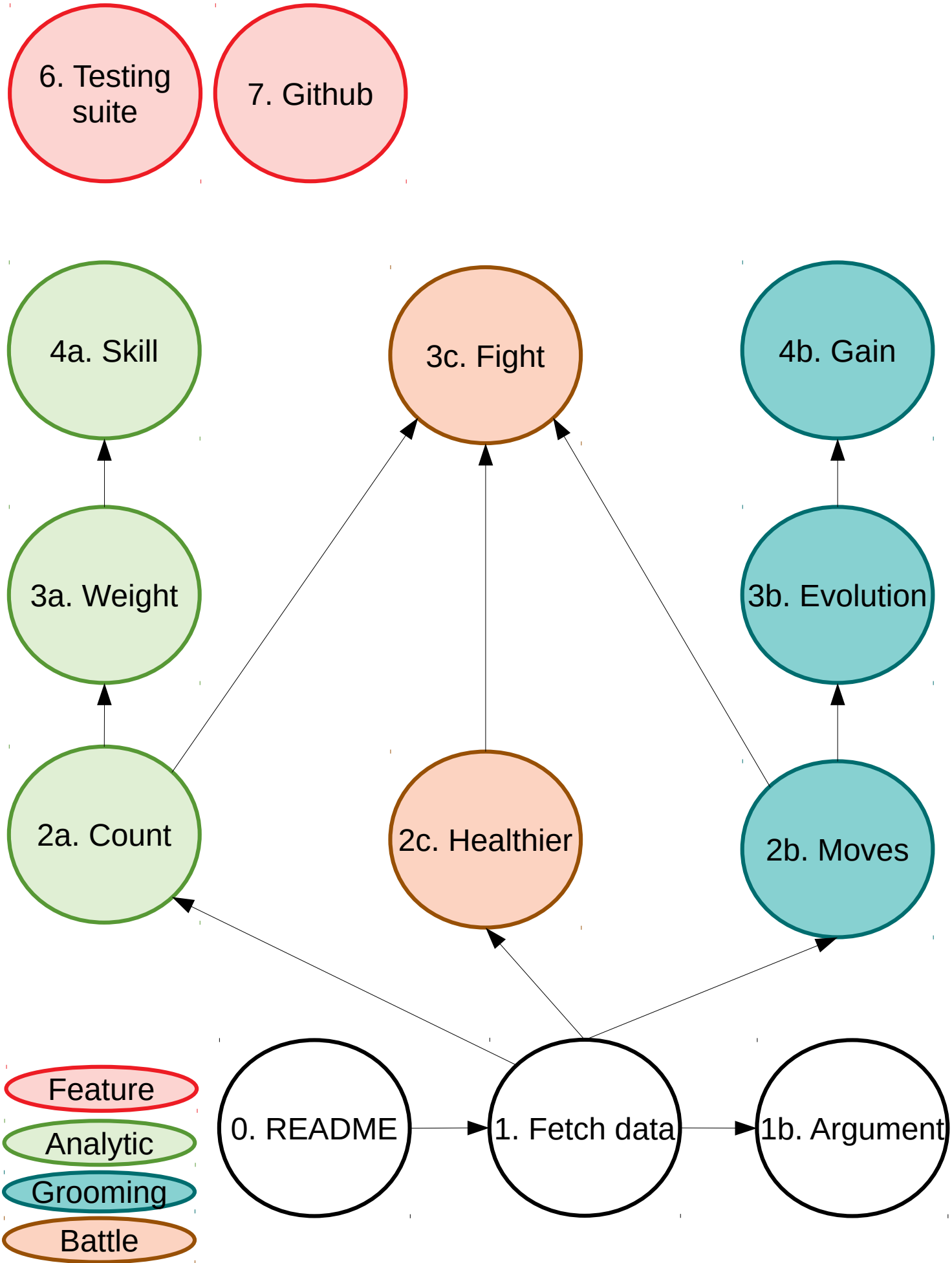
Evaluations:

Live evaluation with teacher	The program needs to work. Students need to prove they succeed the 100 points in front of the teacher.	50%
Code base	Good variables name, use of functions and imports. Respect the presented format in the examples.	50%

Submission

Submit as a ZIP file on Léa before the end of May 15th.

Table of Contents


6. Testing suite
7. Github
4a. Skill
3c. Fight
4b. Gain
3a. Weight
3b. Evolution
2a. Count
2c. Healthier
2b. Moves
0. README
1. Fetch data
1b. Argument
Feature
Analytic
Grooming
Battle

0. README (5 points)

Create a folder into which you will create the structure of the project as follow:

1. Create a file named README.md that contains your team number and the members names. Add a line where you will define the path you chose to get the needed score.

Example:

> Team 1
>
> Antoine LeBel
> Catherine Charest
>
> Chosen path: 0, 1, 2a, 2b, 3a, 4a, 6, 7

2. Create a file name main.py. That file will be the entry point of your program and will be executed like so:
python3 main.py

1. Fetch data (15 points)

Using the module “requests” (pip install requests), call the API by using one pokemon name. Put that pokemon into a variable. Provide the HP (Health Point) and weight for that pokemon.

You should be able to provide the following result to get going:

```
python3 main.py
# r = requests.get(“https://pokeapi.co/api/v2/pokemon/jigglypuff”)
-> I choose “Jigglypuff”. Its HP is 115 and its weight 55.
```

Ref

- https://requests.readthedocs.io/en/latest/user/install/#install

1b. Argument (5 points)

Using “sys.args”, pass a pokemon name to the program.

```
python3 main.py jigglypuff
# response = requests.get(“https://pokeapi.co/api/v2/pokemon/” + my_pokemon)
-> I choose “Jigglypuff”. Its HP is 115 and its weight 55.
```

Ref

- https://www.geeksforgeeks.org/how-to-use-sys-argv-in-python/

- https://docs.python.org/3/library/sys.html#sys.argv

2a. Count (5 points)

Using the API find out how many pokemons exist.

3a. Weight (20 points)

Scanning all pokemons, find out using the API how many pokemons weight more than the chosen pokemon. Different strategies can be used here.

Ref

- https://www.w3schools.com/python/ref_func_len.asp

- https://www.w3schools.com/python/ref_list_append.asp

4a. Skill (30 points)

Looking at the first ability of your pokemon, list all pokemons which have the same ability.

```
python3 main.py jigglypuff
-> Cute Charm ability is possessed by: Clefairy, Clefable, Wigglypuff, Cleffa...
```

*Be sure to list **all pokemons separated with a comma** except for the chosen one.*

3b. Moves (15 points)

Get a list of the first 5 moves the chosen pokemon has.

```
python3 main.py squirtle
Squirtle has the followings moves:
1. Mega-Punch
2. Ice-Punch
3. Mega-kick
4. Headbutt
5. Tackle
```

3b. Evolution (15 points)

Find out if the choosen pokemon can evolve.

```
python3 main.py ditto
-> Ditto does not evolve.

python3 main.py pidgey
-> Pidgey evolves to Pidgeot.
```

4b. Gain (30 points)

List the benefits of the given evolution. If there is no evolution, nothing should appear.

```
python3 main.py pidgey
-> Pidgey evolves to Pidgeot
-> HP increased by 43, height by 12 and weight by 377.
```

Note that it's the difference between the old and the new values.

2c. Healthier (15 points)

Randomly look up a pokemon and checks if its HP is higher than the choosen pokemon. If it is, it wins, if not, it loses.

```
python3 main.py pikachu
-> Pikachu fought for its health against Pyukumiku and lost!
python3 main.py charizard
-> Charizard fought for its health against Pyukumiku and won!
```

Ref

- https://docs.python.org/3/library/random.html

- https://www.w3schools.com/python/ref_func_len.asp

3c. Fight (55 points)

It's time to fight against an other pokemon. Each pokemon possess moves that have a power value. Randomly choose attacks between at least the 5 first moves and reduce the HP according to the attack. Some moves don't have any power, in this case it does nothing.

```
python3 main.py pikachu
-> Pikachu will now fight against Charizard
-> Pikachu attacks Charizard with Thunder Punch.
-> Charizard loses 75HP. It now has 3HP left.
-> Charizard attacks Pikachu with Sword Dance.
-> It does nothing.
-> Pikachu attacks Charizard with Slam.
-> Charizard loses 80HP.
-> Pikachu wins!
```

Ref

- https://docs.python.org/3/library/random.html

- https://www.w3schools.com/python/python_while_loops.asp

6. Testing suite (20 points)

Create five tests that can be launch using "pytest". These five tests need to be on functions and succeed.

Ref

- https://docs.pytest.org/en/7.2.x/getting-started.html

7. Github (30 points)

Create a github repository into which the project will exist. The repository has to have at least one commit from each member.