

Rapport de Projet — SAÉ 1.05 (PITOIS Bastian et RAVELOSON Ho Koloina / TP1.2)

1. Choix effectués

Statistiques par habitat :

Nous avons choisi d'analyser les habitats des Pokémon en se basant sur divers critères statistiques. Lorsqu'un ID d'habitat est entré, notre code renvoie les informations suivantes :

- Le nombre de Pokémon dans l'habitat
- Les noms et images des Pokémon
- Des statistiques moyennes : PV (Points de Vie) moyens, chance moyenne de capture, attaque moyenne, défense moyenne, vitesse moyenne.

Types de Pokémon : Liste des types existants dans l'habitat, avec le nombre de types uniques présents

Nous avons sélectionné ces statistiques pour démontrer comment la puissance des Pokémon varie selon leurs habitats et pour offrir une compréhension approfondie de la diversité et des caractéristiques des Pokémon dans différents environnements.

b) Choix techniques

Pour les choix techniques, nous avons suivi les instructions du professeur en utilisant les bibliothèques suivantes : requests, markdown, json, etc. Nous avons séparé les fonctionnalités en trois fichiers :

- `md_to_html.py` : conversion de Markdown en HTML.
- `pokefiche.py` : génération de fiches Pokémon.
- `pokestats.py` : statistiques des Pokémon selon l'habitat.

Nous avons rencontré quelques difficultés lors de la réalisation, ce qui nous a conduit à ajouter d'autres éléments que ceux initialement demandés, que vous verrez dans la section suivante.

2. Difficultés rencontrées et solutions trouvées

a) Difficultés

1. Lenteur des requêtes API :

- Chaque requête API était lente, surtout pour récupérer plusieurs Pokémon, ce qui ralentissait notre projet pour voir le résultat de nos commandes.

2. Organisation des données JSON :

- Il était complexe de naviguer dans les données JSON retournées par l'API, et nous nous perdions souvent pour trouver les informations désirées.

3. Gestion des erreurs :

- Certaines erreurs empêchaient nos requêtes de fonctionner exemple en lançant le programme tel qu'il était écrit `python3 pokefiche.py <id_pokemon>`.
- Ou l'identifiant était inexistant pour une recherche d'information sur un habitat, compris entre 1 et 9.

4. Traductions :

- La traduction était longue et nous nous perdions parfois dans ce processus.

b) Solutions

Nous n'avons pas trouvé des solutions à toutes nos difficultés, mais nous avons fait de notre mieux. Par exemple :

- Correction du problème avec les ID en affichant un message d'erreur lorsque la limite est dépassée.
- Exploration plus attentive de l'API et réalisation d'essais réguliers pour éviter les erreurs.
- Correction de l'erreur lors du lancement du code avec la commande `python3 pokefiche.py <id_pokemon>`. Remplacez-la par `python .\pokefiche.py` ou `python .\pokestats.py`.
- Utilisation de DOC python et de vidéos youtube exemple : <https://www.python.org/doc/>

3. Apprentissages

Ce projet nous a beaucoup aidé à améliorer nos connaissances et nos pratiques en renforçant notre compréhension des bonnes pratiques en programmation. Nous nous sommes amusés à explorer l'API des Pokémon, ce qui a rendu le projet plus divertissant. Nous avons appris à utiliser la bibliothèque markdown pour automatiser la génération de pages HTML. De plus, notre savoir-être s'est amélioré, car nous avons appris à être très attentifs, curieux et patients tout au long du projet.

4. État d'avancement du projet

Malheureusement, nous n'avons pas complètement terminé le projet. Nous avons rencontré des problèmes lors des traductions, qui ne sont pas complètes, ainsi que des difficultés avec la mise en cache, qui s'est avérée plus complexe que prévu et que nous n'avons pas réussi à implémenter correctement.