

# Rapport

13 Mai 2019

## Introduction

Ce projet a été réalisé par Aghiles Ferrat ([ferrat.aghiles@etu.u-pec.fr](mailto:ferrat.aghiles@etu.u-pec.fr)) et Brandon Alves ([alves.da-silva-alves@etu.u-pec.fr](mailto:alves.da-silva-alves@etu.u-pec.fr)). Il a été initié en février 2019 et est toujours en cours de développement. Le but de cette application est de reproduire une version du jeu Pokémon. Pour le développement de ce jeu, nous avons utilisé le framework CodeIgniter. De plus nous avons utilisé la plate-forme de développement WampServer. Enfin l'application est disponible sur GitHub : <https://github.com/Sevla8/Pokémon>.

## Configuration

On se place dans le répertoire père `Pokémon/`.

Le fichier générant la base de données de l'application est `database/Pokémon.sql`.

Le fichier de configuration de la base de données est `application/config/database.php` dans lequel les lignes 78 à 81 sont à modifier en fonction de votre environnement.

Le fichier de configuration principal est `application/config/config.php`, dans lequel on trouve les variables de configurations `$config['base_url']` et `$config['index_page']`.

Le routage de l'application est le suivant :

`$config['base_url']/$config['index_page']/controller/method/`.

Le fichier `.htaccess` permet d'activer le module de réécriture d'URL afin de se passer de `$config['index_page']`.

Enfin les adresses renseignés dans les fichiers JavaScript (`application/links/js/*`) sont écrites en dur. Il faut donc les récrire en remplaçant `http://localhost/Pokémon` par la valeur de votre variable `$config['base_url']`. Une perspective d'évolution serait de contourner ce désagrément en faisant des fichiers PHP plutôt que JS afin d'utiliser la fonction `site_url` qui prends en compte les valeurs de `$config['base_url']` et `$config['index_page']`.

## Inscription

A partir d'ici, on ne renseignera plus que le nom du controller et de sa méthode sous la forme `controller/method` pour écrire une adresse au lieu de réécrire à chaque fois `$config['base_url']/$config['index_page']/controller/method/`.

Pour s'inscrire il faut accéder à `user/inscription`. Une fois inscrit, on reçoit un mail pour valider son inscription. Une fois l'inscription validée, on peut se connecter.

## Connexion

Pour se connecter il faut accéder à `user/connection`.

## Home

On a le choix entre gérer son équipe, aller chasser des Pokémon, combattre, voir le pokédex, ou faire des emplettes.

La date de dernière activité est mise à jour.

On reçoit une alerte si on nous a envoyé un challenge.

## Pokédex

On peut regarder les statistiques des différents Pokémon. Le pokédex considère qu'Evoli n'a pas d'évolution pour simplifier les choses.

C'est ici que votre aide sur « comment sélectionner des données » dans la base de données nous a été indispensable et nous a aidé tout au long du projet.

On peut rechercher un Pokémon dont on connaît le nom.

## Team

On peut gérer son équipe ou gérer le PC (là où sont stockés les Pokémon qui ne font pas partie de l'équipe).

Dans la gestion de l'équipe on peut modifier l'ordre des Pokémon ou utiliser une potion (la potion restaure tout les PV).

Dans la gestion du PC on peut stocker un Pokémon ou en retirer un dépendamment du nombre de Pokémon dans son équipe. On accepte une équipe constituée d'aucun Pokémon.

## Shop

Dans la boutique on peut acheter des potions ou des pokéballs si on a suffisamment de pokédollars.

## Hunt

Ici on tombe sur un Pokémon choisis aléatoirement. Cependant plus l'attribut « rarity » de la table pokédex est élevé et moins on a de chance de tomber sur ce Pokémon. Ici on n'a pas pris en compte la gestion des équipes : on ne peut pas changer de Pokémon en cours de combat et si notre Pokémon est vaincu le Pokémon sauvage s'enfuit. Une attaque qui n'inflige pas de dégât, et donc qui est censé altérer les statistiques des Pokémon sont prises en compte mais n'ont aucun effet. Par exemple « poudre dodo » n'endort pas l'adversaire et « flash » n'augmente pas la vitesse de l'attaquant. A tout moment on a la possibilité de lancer une pokéball. Si le Pokémon sauvage a moins de 10% de ses PV alors on a une chance sur deux de l'attraper. Si c'est le cas on est redirigé vers la gestion de l'équipe.

## Fight

Ici on peut défier un adversaire connecté, accepter un challenge, envoyer un challenge à un adversaire connecté ou combattre. Si on accepte un challenge on est redirigé vers le combat. On peut alors changer de Pokémon, utiliser une potion, ou attaquer. À chaque tour les Pokémon qui combattent gagnent 10 points d'XP. Ils peuvent également monter en niveau et évoluer. Lors d'une montée en niveau ou d'une évolution les Pokémon apprennent automatiquement leurs attaques. A la fin d'un combat (lorsqu'un entraîneur n'a plus de Pokémon en état de combattre), le gagnant gagne 100 pokédollars et le perdant en perd 100. Le jeu accepte de faire crédit au perdant s'il n'a pas suffisamment de pokédollars, en attendant que ce dernier se refasse.

Une perspective d'évolution serait de faire un chronomètre pour gérer les cas d'abandons : au bout d'un certain temps d'inactivité le joueur dont c'est le tour est déclaré perdant.

En relisant le sujet nous avons compris que les deux joueurs doivent jouer en même temps : un tour équivaut à une action d'une part et d'autre. Nous avons créé une nouvelle branche intitulée **fight** pour traiter les combats de cette manière mais nous nous sommes heurtés à des difficultés et avons préféré continuer de la manière dont nous avons commencé par peur de ne pas réussir à terminer les combats avant la date fatidique. En effet ici un tour équivaut à une action d'un joueur.

Nous n'avons pas non plus considéré la vitesse d'attaque des Pokémon.

## Conclusion

Ce projet nous a été particulièrement instructif dans l'utilisation du modèle MVC ainsi que dans l'utilisation du framework CodeIgniter. Cependant il est possible que le fait d'utiliser un framework nous ait un peu trop facilité les choses. A refaire, nous n'aurions pas utilisé les bibliothèques et les fonctions de CodeIgniter afin de pratiquer d'une manière plus conséquente les bases.