

Monstres de poche

1 Intro, Dates et modalités

Le but de ce projet est de créer un site web qui permet de jouer à un jeu Pokémon, basique (sans la partie histoire/déplacement du personnage). Je vais donner les règles de ce jeu, mais vous êtes libre de les modifier/améliorer. Autant que possible, le projet est incrémental dans sa difficulté.

La notation se fera à la suite d'une soutenance. Le projet se fait par groupe de deux, ou seul. À noter qu'il est vigoureusement conseillé de se mettre avec quelqu'un de niveau comparable, c'est bien plus profitable pour les deux, agréable et la notation sera plus juste. Même si vous répartissez l'écriture du code, vous devez être capable de tout expliquer lors de la soutenance. Dans le cas contraire, cela sera sanctionné sans pitié (la note se fait par personne et non par binôme).

Il est conseillé d'utiliser git et de pusher régulièrement, que vous soyez seul ou à deux. Il y aura des séances spécialement dédiées à l'avancement du projet, et afin que je puisse répondre aux questions.

Le projet est à rendre pour le **mercredi 15 Mai 2018** (juste avant le début du cours). Le rendu se fait par la soumission sur le site wim.choum.net de l'ensemble de votre code, ainsi qu'un fichier au format `.sql` contenant une base de donnée, contenant toutes les tables dont vous avez besoin pour votre projet¹. **Vous devez aussi envoyer un court rapport** (1 à 2 pages), présentant votre projet : ce que vous avez réussi à faire, les problèmes que vous avez éventuellement eu, toute remarque que vous voulez faire. Le rapport est là pour m'aider à comprendre le projet, son fonctionnement et sa création, ainsi qu'à noter le projet.

Vous devez m'indiquer votre groupe de 2 ou de 1 au plus tard **mercredi 13 Mars 2018**, par e-mail (peada@free.fr). Les notes ne seront pas par groupe, mais par personne.

Les soutenances auront lieu le **mercredi 22 Mai 2018**, d'une durée de 15 à 20 minutes.

2 Description du jeu Pokémon

Le but est d'implémenter une version simplifiée du jeu de pokémon, contenant une version simplifiée des combats et une manière plus ou moins automatique de chasser de nouveaux pokémons. Cependant, bien que tout soit simplifié, cela n'en reste pas moins un projet très compliqué ! N'essayez donc pas d'implémenter tout dès le début (cf les différentes évolutions).

Pour ceux qui ne connaissent pas pokémon, voici un simulateur de combats de pokémons : <https://play.pokemonshowdown.com/>. Et voici la description du jeu à implémenter, dans sa version complète.

2.1 Création d'un compte

Lorsque l'on crée un compte, on a le choix entre trois pokémons (par exemple Salamèche, Bulbizarre et Carapuce). Ce pokémon constituera notre premier pokémon.

On peut ensuite se connecter en utilisant le login et mot de passe choisi à la création de compte. Le tout doit être sécurisé.

2.2 Nouvelle journée

Au début de chaque journée (heure française), chaque utilisateur inscrit récupère certaines ressources :

- Tous ses pokémons récupèrent leur santé,
- Il reçoit 50 P(pokédollar) et 5 pokéballs.

1. Pour obtenir ce fichier `.sql`, utilisez la fonctionnalité d'export de phpmyadmin.

2.3 Actions possibles

A tout moment, un utilisateur connecté peut faire les actions suivantes :

- Gérer son équipe de pokémon,
- Aller chasser des pokémons sauvages.
- Acheter des pokéball ou des potions de soin.
- Combattre contre un autre dresseur de pokémons connecté.

2.4 Gérer son équipe de pokémon

Une équipe de pokémon est constituée d'au moins un pokémon, et d'au maximum six pokémons. Un dresseur peut cependant posséder plus que 6 pokémons.

Le dresseur peut rajouter ou enlever des pokémons à son équipe, ainsi que choisir le premier pokémon à combattre.

2.5 Chercher des pokémons

Lorsque vous allez chercher des pokémons, vous avez une chance de trouver un pokémon aléatoire, et d'engager un combat contre lui. Plus le pokémon est puissant, moins vous avez de chance de le trouver.

À tout moment du combat, vous pouvez tenter de capturer un pokémon en lançant une pokéball, et vous réussissez s'il a suffisamment peu de vie.

2.6 Combats

Les combats se déroulent de la manière suivante :

- Au début de chaque tour, les deux opposants choisissent soit une attaque, soit de changer de pokémon en en choisissant un parmi ceux de l'équipe encore vivant.
- En cas de changement de pokémon, cela a lieu en premier.
- Ensuite, le pokémon le plus rapide attaque en premier. On applique les dégâts, en fonction du niveau du pokémon et des éléments des attaques et du pokémon attaqué.
- Le pokémon le plus lent effectue aussi son attaque s'il lui reste des points de vie. Le tour suivant peut commencer.

2.7 Combat contre un joueur

Dans l'interface, vous devez pouvoir voir la liste des joueurs connectés et les défier en duel. Si vous défiez un joueur, il a une notification et peut accepter le combat. Celui-ci démarre alors, et le gagnant vole de l'argent au perdant.

3 Codez !

La manière dont sera structurée votre projet est très importante, elle peut le rendre beaucoup trop complexe ou au contraire très simple à programmer ! Réfléchissez donc bien à vos choix.

Vous avez le choix entre utiliser <https://www.codeigniter.com/> ou structurer vos fichiers vous même. Dans tous les cas, factorisez au maximum votre code.

Par exemple, (sans codeigniter) vous devriez utiliser au maximum les `include` afin de réutiliser les fonctions. Par exemple :

- Ayez un fichier `security.php` qui vérifie que l'utilisateur est bien connecté, et le redirige vers la page de login si besoin. Incluez-le dans toutes les pages nécessitant d'être connecté.
- Ayez un fichier `bdd.php`, et un seul, contenant ce qu'il faut pour avoir accès à la base de donnée,
- Ayez des fichiers `pokemons.php` et `users.php` contenant toutes les fonctions utiles en lien avec les pokémons, et les utilisateurs.

Exercice I (Gestion des utilisateurs)

Implémentez tout ce qu'il faut pour créer un compte et se connecter : les formulaires d'inscription et de connexion, l'ajout à la base de donnée d'un nouvel utilisateur. Faites quelque chose d'aussi sécurisé que possible, en particulier, cryptez le mot de passe !

Exercice II (Jeu, évolution embryon)

Implémentez les fonctionnalités de base du jeu dans une version très simplifiée :

- Les joueurs ont argent infini,
- Lors d'une chasse au pokémon, on gagne un pokémon sans combattre,
- Pas de combat entre joueurs...

Cependant, faites les pages `html` de manière complète. Essayez aussi d'avoir une base de donnée qui vous permet de rajouter les fonctionnalités non simplifiées (par exemple, même si les pokémon ne peuvent pas encore perdre de points de vie, ajoutez dès à présent un champs dans la table `pokemon`).

Exercice III (Jeu, évolution enfant)

Implémentez les fonctionnalités :

1. Gain de points de vie et argent chaque nouveaux jour,
2. Achat et utilisation de potions hors combats, achat de pokéball
3. Les pokémons perdent des points de vie lors des combats (qui restent simplifiés), capturer un pokémon consomme une pokéball
4. Ajouter la gestion des équipe.

Exercice IV (Jeu, évolution adulte)

Implémentez les fonctionnalités :

1. Les combats contre les pokémons sauvages ne sont plus simplifiés.
2. On doit utiliser une pokéball pour capturer un pokémon s'il a peu de vie.

Exercice V (Jeu, méga-évolution)

Implémentez les fonctionnalités :

1. Les combats contre les autres dresseurs (pour cela, vous aurez besoin d'AJAX).
2. Les pokémon gagnent de l'XP et montent en niveau.
3. Tout ce que vous voulez améliorer, si vous avez déjà tout ça c'est excellent !

4 Notes et conclusion

Ce projet est très dur, ne vous angoissez pas si vous n'arrivez pas au bout. De plus, il n'est pas trop guidé, afin que vous vous creusiez la tête sur la meilleure manière de construire votre code, et que vous fassiez mieux la prochaine fois.

Pour la notation, il sera tenu compte de plusieurs choses :

1. La connaissance de son code lors de la soutenance,
2. la propreté et l'efficacité du code, sa modularité et son extensabilité,
3. la quantité de fonctionnalité implémentée,
4. l'absence de bugs,
5. la sécurité du site : faille XSS, injection de code SQL
6. la sécurité du jeu : on ne doit pas pouvoir tricher !
7. Petit bonus pour l'utilisation de git...

Bon courage!!!