

TD Pokémon

NoSQL



Objectif	2
Modalités de réalisation et de rendu	3
Critères d'évaluation	3
Terminologie	4
Dictionnaire de données	4
Déroulé d'une bataille et d'un combat	5
Consignes	6
Crédits	8
Documentation	8
Données JSON	8

Objectif

L'objectif du projet est de développer une application permettant de :

- générer des équipes de 6 *Pokémon*,
- organiser et faire progresser un tournoi à élimination directe, intégrant a minima 32 équipes,
- simuler des batailles au tour par tour entre équipes de *Pokémon*,
- générer des données correspondant aux résultats obtenus lors des combats et batailles (ex : victoire / défaite de chaque *Pokémon*, nombre de points de vie perdus et perte de points infligés pour chaque combat...),
- calculer et fournir des données statistiques de votre choix au moyen de requêtes sur la base de données (ex: taux de victoire de chaque *Pokémon*, ...).

En raison du temps court alloué au projet, certaines fonctionnalités de l'application pourront être réalisées sous forme de **preuve de concept**.

Vous êtes libres d'adapter le sujet selon votre inspiration (fonctionnalités, logique métier...).

Modalités de réalisation et de rendu

- Travail individuel ou en binôme.
- Travail à rendre sous forme de dépôt *Git* distant (*Github* ou *Gitlab*) partagé à alex@shrp.dev (*shrp777*) au plus tard le 21/2/2025 à 23h59 (aucun délai supplémentaire ne sera accordé).
- 🚨 **Prénom et nom de l'auteur ou des auteurs dans un fichier README.md placé à la racine de votre projet.** 🚨
- Pour la qualité de votre apprentissage, un usage modéré de l'IA générative est recommandé.

Critères d'évaluation

- Qualité du code,
- Qualité conceptuelle,
- Fonctionnalités implémentées,
- Emploi pertinent de *NoSQL*,
- Documentation,
- Participation et comportement pendant les cours.

Terminologie

- **Pokédex** : liste des personnages *Pokémon*.
- **Tournoi** : compétition entre 32 équipes de 6 *Pokémon*.
- **Bataille** : opposition entre 2 équipes de 6 *Pokémon*.
- **Combat** : opposition entre 2 *Pokémon*.

Dictionnaire de données

- **stats.hp** (nombre entier) : points de vie (niveau de résistance de départ de chaque *Pokémon*),
- **stats.atk** (nombre entier) : attaque, dégâts infligés à chaque attaque (exprimés en points de vie),
- **stats.def** (nombre entier) : défense, capacité de protection face à une attaque (réduit d'autant de points de vie les dégâts infligés par l'attaque de l'opposant),
- **stats.vit** (nombre entier) : vitesse (exprimée en points de vitesse).

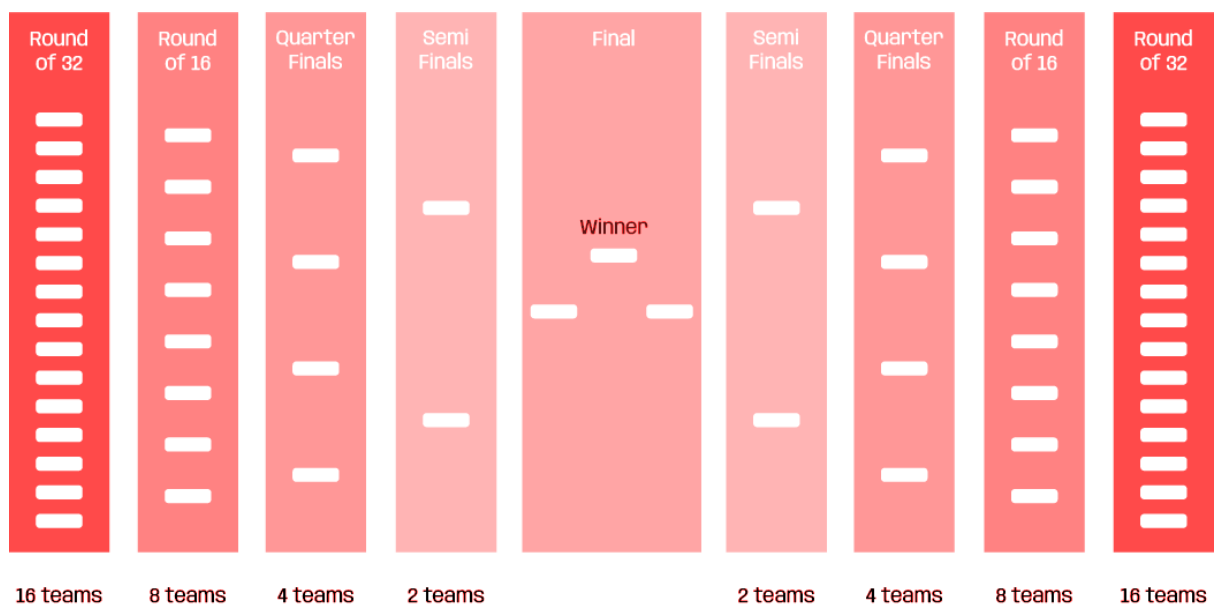
Déroulé d'une bataille et d'un combat

- Chaque équipe contient 6 *Pokémon*,
- Lorsqu'une bataille entre 2 équipes débute, un *Pokémon* de chaque équipe est tiré au sort,
- Chaque *Pokémon* débute le combat avec le nombre de points de vie complet selon la valeur de son attribut **stats.hp**,
- Le *Pokémon* le plus rapide parmi les 2 Pokémon sélectionnés (cf. attribut **stats.vit**) attaque en 1er. Si les 2 valeurs stats.vit sont égales, procédez à une sélection aléatoire.
- Les dégâts infligés au *Pokémon* adverse correspondent à la valeur de l'attribut **stats.atk** de l'assaillant. En option, les dégâts infligés au *Pokémon* adverse peuvent être modérés par la valeur de l'attribut **stats.def** du *Pokémon* visé). **Adaptez librement la mécanique de combat.**
- Un combat 1 vs 1 se conclut lorsque l'un des deux **Pokémon** n'a plus de point de vie,
- Le *Pokémon* vainqueur (avec son niveau de vie en l'état) est opposé au *Pokémon* suivant de l'équipe adverse (tiré au sort), et ainsi de suite, jusqu'à ce que l'une des deux équipes ne dispose plus de *Pokémon* avec des points de vie.
- L'équipe disposant d'au moins 1 *Pokémon* avec des points gagne la bataille et poursuit le tournoi.
- A la bataille suivante, tous les *Pokémon* de l'équipe victorieuse débutent le combat avec leurs points de vie par défaut (cf. attributs **stats.hp**).
- L'équipe perdante est éliminée du tournoi.

Consignes

- Mettez en place une infrastructure locale à l'aide de **Docker** et de **Docker Compose**,
- Définissez un **network Docker** nommé **pokemon_network**,
- Créez un service **Docker** nommé **db** associé à l'image **Docker** de **MongoDB** (ou autre type de Base de données *NoSQL* de votre choix, adapté au projet),
- Au sein du service **Docker db**, créez une base de données nommée **pokemon**,
- Importez les données **JSON** fournies dans une collection nommée **pokedex** (cf. fichier *pokemon.json*).
- Mettez en place un service **Docker** nommé **app** permettant à une application de s'exécuter dans un container **Docker** (stack technologique votre choix).
- Programmez une application permettant :
 - d'accéder aux données de la collection nommée **pokedex**, afin de récupérer les données des personnages **Pokémon**.
 - constituer de façon aléatoire 32 équipes de 6 *Pokémon* (un même *Pokémon* ne peut pas appartenir à plusieurs équipes participant au tournoi).
 - enregistrer les 32 équipes générées,
 - organiser un tournoi regroupant les 32 équipes générées aléatoirement, et le faire progresser depuis les 16èmes de finale jusqu'à la finale,
 - générer des combats 1 vs 1 au tour par tour entre équipes de 6 *Pokémon*,
 - stocker les données relatives aux batailles et combat du tournoi,
 - calculer et fournir des **statistiques** de votre choix basées sur les données générées au cours du tournoi.
- Selon la stack technologique sélectionnée, employez (si possible) un **ODM** afin de faciliter vos interactions avec la base de données (ex: **Mongoose** pour *JS / TS*, **Beanie** pour *Python*, **Doctrine** pour *PHP*, **Mongoid** pour *Ruby*... cf. <https://www.mongodb.com/developer/products/mongodb/mongodb-orms-odms-libraries/>) et définissez les **entités** correspondant aux données manipulées dans le programme,

- Dans la base de données **pokemon**, créez une seconde collection nommée **battles** permettant de stocker toutes les données générées lors des combats :
 - composition des 2 équipes de *Pokémon*,
 - résultat obtenu pour chaque combat 1 vs 1 :
 - points de vie perdus pour chaque *Pokémon*,
 - coups infligés exprimés en point de vie pour chaque *Pokémon*,
 - *Pokémon* vainqueur,
 - équipe victorieuse.



Format du tournoi

Crédits

Documentation

- <https://www.pokemon.com/fr>
- <https://leclaireur.fnac.com/article/257426-on-vous-explique-comment-vraiment-jouer-aux-cartes-pokemon/>
- [https://www.pokepedia.fr/Liste des Pok%C3%A9mon dans l'ordre du Pok%C3%A9dex_National](https://www.pokepedia.fr/Liste_des_Pok%C3%A9mon_dans_l'ordre_du_Pok%C3%A9dex_National)

Données JSON

- <https://github.com/Yarkis01/TyraDex/blob/main/data/pokemon/pokemon.json>