

Projet 2 – Collectible Card Game

Encadrant : Guillaume Hivert

Etudiants : MEDDAH Sarah, MARTINEZ Luis

Table des matières

1	Introduction	2
2	Partie on-chain :	2
3	Partie off-chain :	4
4	Déploiement et Configuration :	5
5	Pour aller plus loin :	5
6	Conclusion :	6

1 Introduction

L'objectif de ce projet est de créer une application décentralisée (dApp) qui permet aux utilisateurs d'interagir avec une collection de cartes basée sur la blockchain. Les utilisateurs peuvent créer des collections les échanger (les vendre, ou en acheter d'autres) sur une plateforme dédiée. Le projet tire parti des smart contracts Ethereum pour garantir des transactions sécurisées et transparentes. Grâce à cette infrastructure, nous offrons une expérience interactive aux amateurs de TCG, tout en assurant l'authenticité et la traçabilité de chaque élément de collection puisque Ethereum offre un environnement idéal pour la création de cartes de collection sous forme de NFT unique et traçable sur la blockchain, garantissant cette authenticité et cette rareté des cartes de collection.

Les principaux étapes du Projet :

Compréhension des composants du jeu :

- Identification des éléments on-chain (hébergés sur la blockchain) et off-chain (frontend et backend).
- Création d'un cadre pour gérer les cartes numériques et permettre leur échange.

Développement de l'infrastructure :

- Mise en œuvre d'un contrat NFT conforme à la norme ERC-721.
- Création d'un frontend pour interagir avec le contrat et visualiser les cartes.

Intégration de l'API Pokémon TCG :

- Ajout de vraies cartes avec illustrations récupérées de l'Api Pokémon.
- Modification des API pour intégrer les données des cartes Pokémon.

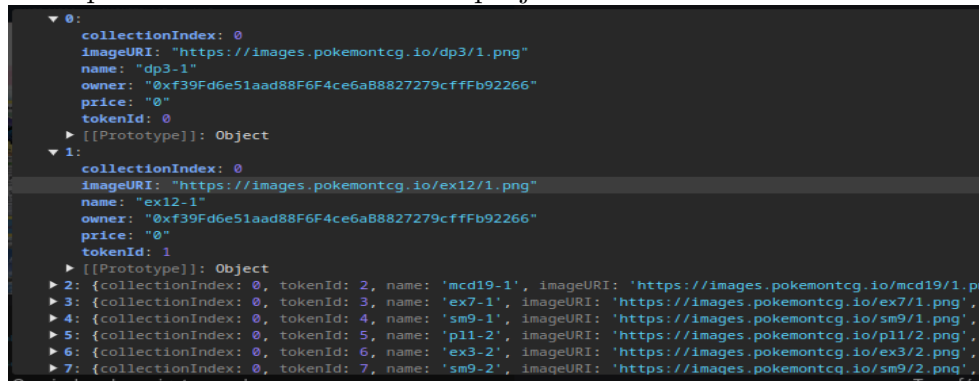
2 Partie on-chain :

Création de contrat NFT :

Le contrat NFT constitue la pierre angulaire de notre projet, permettant la création et la gestion de collections de cartes, chacune définie par un nom et un nombre de cartes. Grâce à sa fonctionnalité de création de collections, il est possible d'établir de nouveaux ensembles de cartes qui peuvent être frappés en tant que NFT. Le contrat principal, baptisé **Main.sol**, s'intègre dans l'environnement de développement **Hardhat**, facilitant ainsi le déploiement et la gestion des contrats intelligents. Ce contrat gère non seulement la création de collections et le minting des cartes, mais il supervise également les opérations du marché, telles que l'achat et la vente des cartes. chaque appel de fonction débute au sein du contrat Main, qui s'assure ensuite de communiquer avec le contrat Collection

pour réaliser les opérations requises. Cette approche garantit la cohérence des données tout en renforçant la sécurité globale du système, créant ainsi une plateforme robuste pour l'échange et la gestion des cartes.

Exemple de contrat créer dans ce projet :



Implémentation de la norme ERC-721 La norme ERC-721 est utilisée pour représenter les NFT, garantissant leur unicité. Chaque carte contiendra des métadonnées, notamment un numéro de carte et une image. Ce format assure la traçabilité et l'authenticité des cartes sur la blockchain

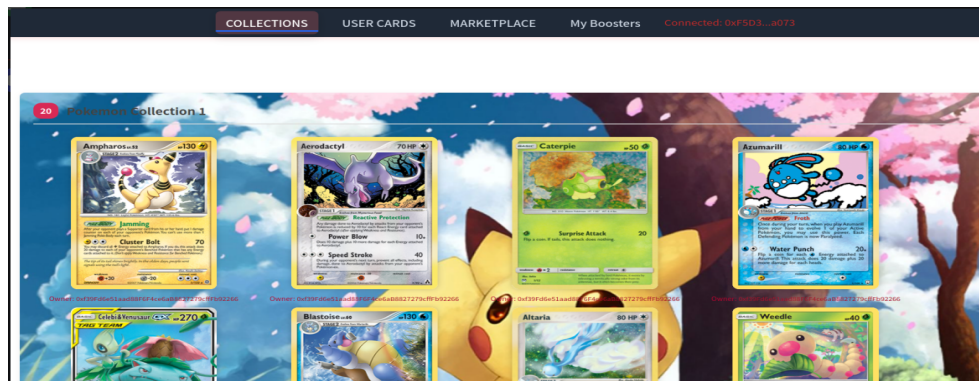
Implémentaion du projet :

Le contrat principal, nommé Main, constitue le cœur de notre système. Il supervise la gestion des collections, les échanges de cartes, les listes de cartes à vendre, ainsi que les informations générales sur l'ensemble du jeu. L'interaction entre les contrats s'effectue de la manière suivante : chaque appel de fonction est d'abord dirigé vers le contrat Main, qui se charge ensuite de communiquer avec le contrat Collection pour réaliser les opérations requises. Cette méthode assure la cohérence des données et la sécurité de l'ensemble du système.

Création de Collections

Dans notre implémentation, les collections sont créées immédiatement après le déploiement du contrat. Pour créer une nouvelle collection, une fonction sur le contrat principal Main est appelée en précisant le nom de la collection et le nombre de cartes souhaité directement dans le code. Cette fonction crée une nouvelle collection sur la blockchain avec les caractéristiques spécifiées. Nous avons utilisé les wallets prédéfinis de la configuration Hardhat. Les collections sont stockées à l'aide d'une variable collections : Cette variable est une sorte de mapping. On peut aussi récupérer toute la collection ou les collections d'un user.

Image de la collection créer :



Backend

Le backend de notre projet, construit avec Express.js, joue un rôle central en tant que couche API entre le frontend et la blockchain. Il est chargé de la récupération des informations sur les cartes NFT en interagissant avec le contrat principal, Main. le backend peut également s'intégrer à des sources externes qui fournissent des métadonnées détaillées sur chaque carte. Ces métadonnées incluent des informations telles que le nom de la carte, sa description, son histoire, etc. Cette intégration garantit une expérience de jeu riche et informative pour les joueurs

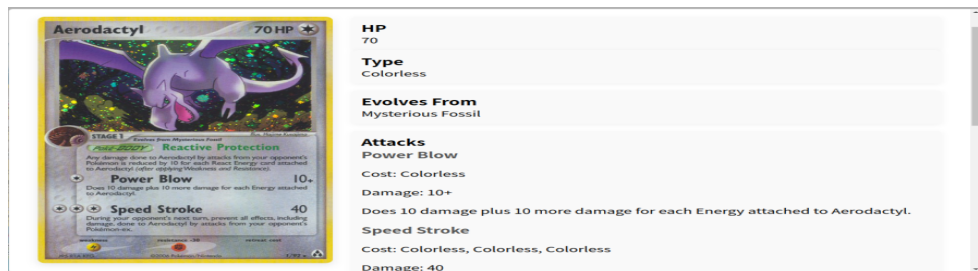
3 Partie off-chain :

Frontend

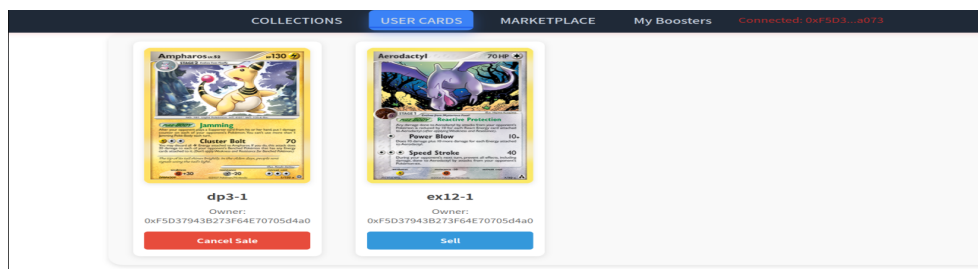
Le frontend est en rapport avec le Backend et utilise les contrats créer, Son rôle d'interface utilisateur est de permettre aux joueurs d'explorer, de collectionner et d'échanger des cartes NFT sur la blockchain Ethereum. Il permet spécialement ces derniers à :

- la visualisation des collections :Les joueurs peuvent voir toutes les collections existantes, y compris les NFT qu'ils possèdent et celles qui sont mises en vente sur le marché avec les informations sur le propriétaire. Les informations sur chaque collection incluent son nom et le nombre de cartes qu'elle contient.
- visualiser les détails des NFTs. Ils peuvent explorer chaque carte NFT pour en savoir plus sur ses caractéristiques, ses capacités, ses effets. Les informations sur chaque carte sont fournies par le backend, qui récupère des métadonnées détaillées, telles que la description et les attributs spécifiques de chaque carte, auprès de sources externes.
- Le frontend affiche également l'adresse du portefeuille connecté, permettant aux joueurs de vérifier leur identité et de s'assurer qu'ils utilisent le bon portefeuille pour effectuer des transactions.

Exemple de détail d'une carte :



La liste des cartes appartenant à l'utilisateur :



Interaction Utilisateur :

Les utilisateurs peuvent connecter leurs portefeuilles Ethereum (par exemple, MetaMask) pour interagir avec l'application.

L'application vérifie si l'utilisateur est sur le bon réseau (réseau local Hardhat) et propose des fonctionnalités pour visualiser et gérer ses collections de cartes et participer au marché.

4 Déploiement et Configuration :

Le fichier `hardhat.config.ts` configure l'environnement Hardhat, en spécifiant les paramètres du compilateur Solidity, les chemins de déploiement et les comptes nommés pour le déploiement. Le projet utilise plusieurs plugins Hardhat pour une fonctionnalité améliorée, tels que `hardhat-deploy` pour les scripts de déploiement, `hardhat-gas-reporter` pour les rapports d'utilisation du gaz, et `hardhat-abi-exporter` pour l'exportation des ABI des contrats.

5 Pour aller plus loin :

Le Marketplace :

Ce marché décentralisé où les joueurs peuvent acheter, vendre et échanger des cartes avec d'autres passionnés. Ce marketplace, fondé sur les fonctionnalités des contrats Main et Collection, permet aux joueurs d'effectuer plusieurs actions essentielles :

Le Marketplace :



Vente des cartes :

Si on veut vendre une carte on doit changer un attribut "onsell" vers true ce qui veut dire qu'elle est mise en vente et peut partir d'un moment à un autre et elle ne fait plus partie de notre possession sauf si je change d'avis et je lui enlève cet état de vente avant qu'elle ne soit vendu.

Achat de cartes :

Les joueurs peuvent parcourir les cartes disponibles sur le marché, visualiser les détails de chaque carte, et acheter celles qui les intéressent. Lorsqu'un achat est effectué, le contrat Main gère la transaction en transférant la carte du vendeur à l'acheteur et en effectuant la transaction en ethers

6 Conclusion :

Ce projet comprend différentes technologies et étapes et pour nous ça été un vrai challenge pour le comprendre, comprendre toutes ses différentes technologies, les relations entre elles et enfin l'implémentation en un temps relativement court par rapport à ce qui est attendu. Le potentiel d'évolution de ce projet est très grand. Faute de temps nous n'avons pas pu faire tout ce qu'on voulait. Des améliorations peuvent être envisagées en termes d'optimisation des contrats et d'interface utilisateur.