SCIENCES SORBONNE UNIVERSITÉ

Projet N°02 :

Collectible Card Game

M2 RES\DEV

KABTANE Rania 21305223
MOUSSOUNI Yaakoub 21315415

I. Introduction

Notre projet « Collectible Card Game » vise à concevoir un jeu de cartes à collectionner (CCG) décentralisé en s'appuyant sur la blockchain Ethereum. L'objectif principal est de permettre aux joueurs de posséder, échanger et collectionner des cartes sous forme de NFTs (Non-Fungible Tokens). Contrairement aux jeux traditionnels où les cartes sont détenues de manière centralisée, ici, chaque joueur détient une preuve de propriété immuable sur la blockchain, rendant les cartes uniques et traçables. Ce projet nécessitera plusieurs étapes, notamment la création des cartes sous forme de NFT, la mise en place d'un marché décentralisé pour faciliter l'échange entre joueurs, et le développement d'une interface conviviale pour que les utilisateurs puissent créer et gérer leurs collections.

Pour répondre à ces besoins, nous avons concentré nos efforts sur trois composants principaux :

1. Backend : gère la logique serveur en dehors de la blockchain, comme le suivi des utilisateurs, la gestion des données liées aux cartes, et la coordination des actions des joueurs. Il s'occupe également de l'interaction avec les smart contracts pour exécuter les transactions sur la blockchain Ethereum.
2. Frontend : Interface utilisateur avec laquelle les joueurs interagissent. Il permet aux utilisateurs de consulter leurs cartes, de gérer leurs collections, d'effectuer des échanges, et de participer au jeu. Il se connecte également à des portefeuilles Ethereum comme Metamask pour interagir avec la blockchain. **Notre source d'inspiration pour cette partie était le projet fait par @Alina Novikova et @Shuhan Duan [1].**
3. Smart Contracts sur la Blockchain Ethereum : Programmes déployés sur la blockchain Ethereum qui gèrent la création, la propriété et l'échange des cartes sous forme de NFTs. Ils assurent que les règles du jeu, les transactions, et les échanges sont exécutés de manière sécurisée, transparente et sans intermédiaire [2][3].

II. Implémentation

L'implémentation du contrat principal, nommé **Main**, Le contrat Main centralise la gestion des collections de cartes, le suivi des propriétaires, et le minting de nouvelles cartes sous forme de NFTs. Il facilite la gestion d'un jeu de cartes à collectionner décentralisé en Ethereum, permettant aux utilisateurs de détenir des cartes traçables et uniques.

1. Partie On-Chain : Création du Contrat NFT

Les cartes sont l'élément clé de notre réalisation. Chacune de ces cartes peut représenter un pokémon, une compétence avec des attributs spécifiques. Ces cartes seront créées sous forme de NFTs (Non-Fungible Tokens) pour garantir leur unicité et leur traçabilité sur la blockchain. Pour cela, nous utilisons des standards comme l'ERC-721 (chaque carte est unique). Chaque NFT aura une métadonnée associée (URI), qui renverra à une image et à d'autres attributs, et sera stockée sur la blockchain Ethereum, assurant que personne ne peut les altérer ou les contrefaire.

Fonctionnalités du Contrat Main

1 Gestion des Collections :

- **Création** : createCollection permet de créer et d'enregistrer une nouvelle collection de cartes en spécifiant un identifiant, un nom, et un nombre de cartes. Chaque collection est initialement détenue par le contrat lui-même.

2 Minting de Cartes (NFTs) :

- **Distribution de Cartes** : mintCard permet de créer et d'attribuer une carte NFT à un utilisateur avec des métadonnées spécifiques.
- **Suivi des Propriétaires** : Lorsqu'un utilisateur reçoit une carte pour la première fois, son adresse est ajoutée à la liste des propriétaires uniques.

3 Consultation des Propriétaires et des Balances :

- **Liste des Propriétaires** : getOwners fournit la liste de tous les utilisateurs ayant reçu des cartes.
- **Balance par Collection** : balanceOf retourne le nombre de cartes possédées par chaque utilisateur dans chaque collection.

2. Partie Off-Chain : Création du Frontend et API

Une interface utilisateur permet aux utilisateurs de visualiser leurs cartes. Elle se compose de deux principales fonctionnalités :

- API d'Informations sur les NFT : Création d'un serveur web fournissant les métadonnées pour chaque carte NFT. Cette API retourne un JSON contenant le nom de la carte et une illustration, mise à jour en fonction de l'image associée au NFT.
- Visualisation des NFT des Utilisateurs : Le frontend, après connexion de l'utilisateur, affiche toutes les cartes NFT détenues par l'utilisateur. L'application interroge la blockchain pour récupérer les NFT, puis appelle l'API pour obtenir les détails de chaque carte.
  Pour commencer, nous avons initialisé le projet, puis installé ethers.js pour interagir avec la blockchain et les requêtes API vers notre backend. Nous avons ensuite configuré la connexion à Metamask pour permettre aux utilisateurs de se connecter à leur portefeuille, et nous avons mis en place une fonction connectWallet pour récupérer l'adresse de chaque utilisateur. Une fois l'adresse obtenue, nous interrogeons la blockchain pour récupérer les NFT de l'utilisateur en appelant notre contrat via ethers.js, pour obtenir les identifiants de chaque NFT détenu. Pour chaque identifiant, nous faisons une requête à notre API afin de récupérer les métadonnées associées, que nous stockons ensuite dans l'état de l'application. Enfin, nous avons créé un composant Card pour afficher chaque NFT, ce qui permet aux utilisateurs de voir leurs cartes dans une galerie avec toutes les informations associées

3. Intégration de l'API Pokémon TCG

Pour enrichir le contenu des cartes, nous avons intégré l'API Pokémon TCG, ce qui nous a permis d'ajouter des cartes populaires et reconnaissables. Nous avons d'abord sélectionné des sets spécifiques de Pokémon TCG et les avons ajoutés au Main contract, en veillant à indiquer le nombre exact de cartes pour chaque set. Ensuite, nous avons modifié notre API pour inclure des informations détaillées sur les cartes Pokémon associées aux NFT, offrant ainsi des illustrations et des détails de collection qui renforcent l'immersion des joueurs dans l'univers du jeu de cartes.

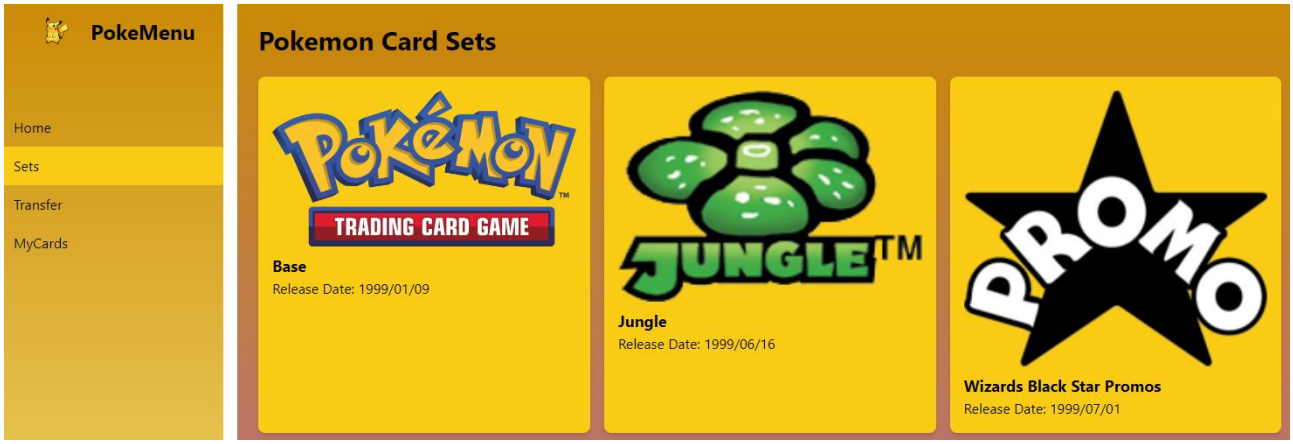

Figure 1 : Pokémon Sets importés via API

L'intégration des sets Pokémon comme précisé auparavant se fait via l'API pokémon [4] mise à disposition.

Son intégration est simple, un fichier JavaScript en backend nous permet de faire appel à l'api grâce à son lien et notre API-key personnelle.

```
backend > JS index.js > ...
1   const express = require('express');
2   const app = express();
3   const port = 3000;
4   const initServer = require('./controllers/initServer');
5   const POKEMONURL = 'https://api.pokemontcg.io/v2';
6
7   const API_KEY = '8b467d29-299a-42ad-a0e5-da506f3e01ec';
8
9   module.exports = {
10      POKEMONURL,
11      API_KEY
12  };
13
14  initServer(app);
15
16  app.listen(port, () => {
17      console.log(`Backend listening at http://localhost:${port}`);
18  }
19  );
20
21
```

Figure 2 : Script Index.js appelant l'API

Ce script nous permet de définir le port d'écoute en local, ainsi que nous authentifier avec l'API pour pouvoir agir dessus librement sans limitations.

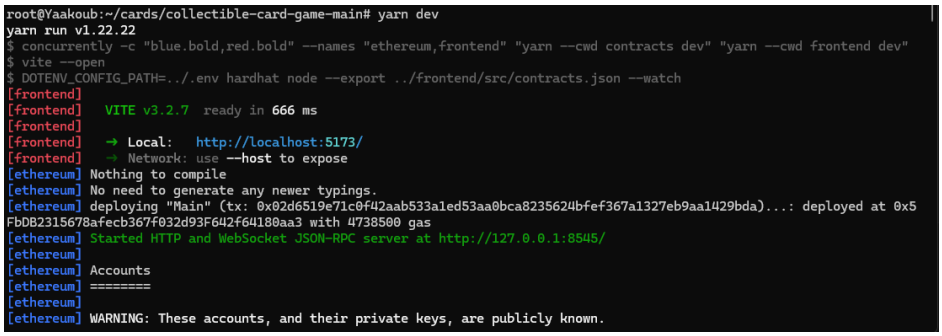

Figure 3 : Lancement de notre projet

In ne suffit que de lancer l'écoute après avoir lancé notre programme principal à l'aide de yarn dev.

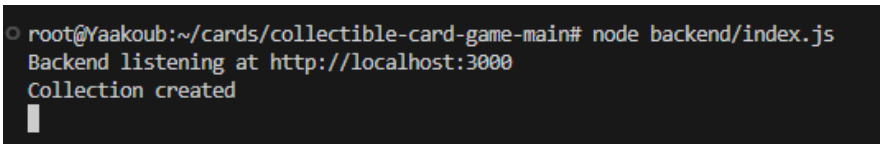

Figure 4 : Ouverture du port d'écoute permettant l'interaction avec l'API

Après cela, se diriger vers http://localhost:5173/

Nous nous retrouvons à la page d'accueil, où nous avons notre Menu, qui nous permet de visiter et de voir les Sets de cartes disponible comme vu dans les figures précédentes, choisir de transférer ou de minter des cartes pour soi, ainsi que de voir nos cartes déjà mint ou l'état de notre panier en attente comme nous le verrons sur les figures qui suivent.

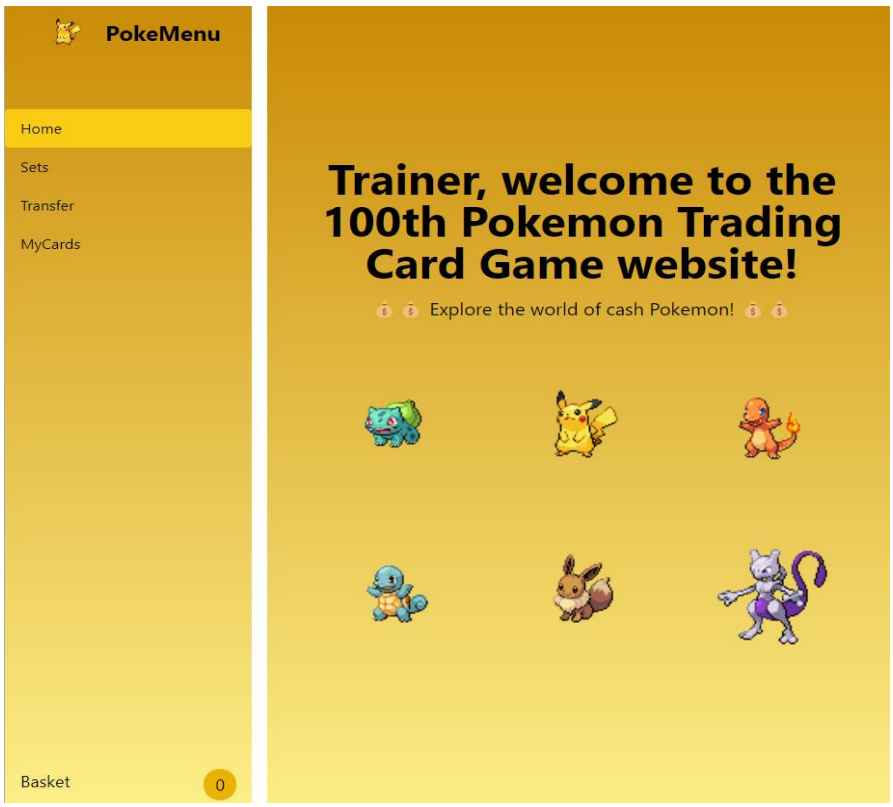

Figure 5 : Accueil de notre site web TCG

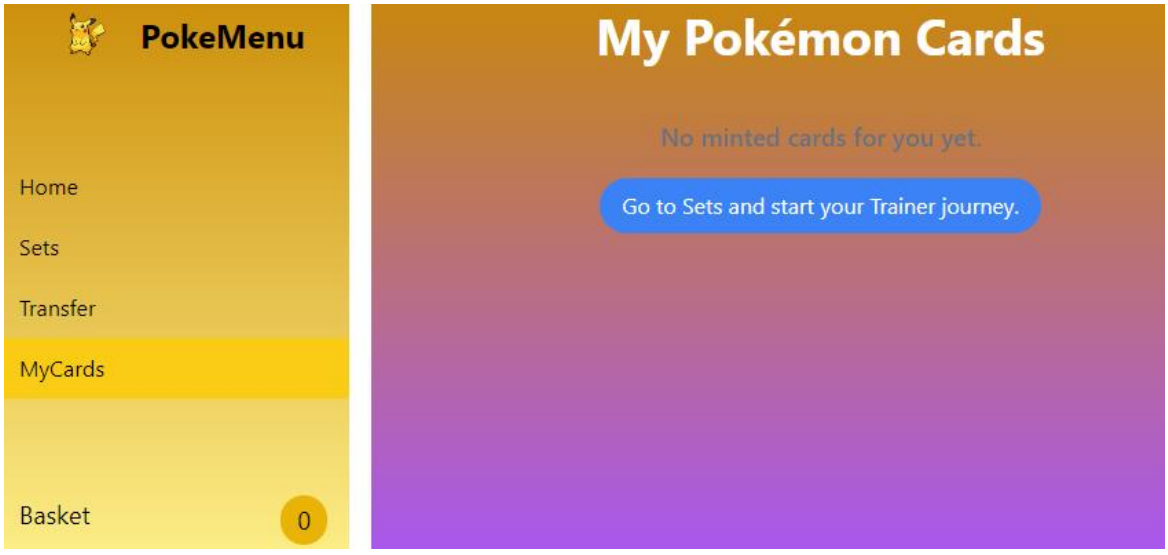

Figure 6 : Inventaire des cartes lié à notre compte Metamask

Figure 7 : Choix des cartes à minter

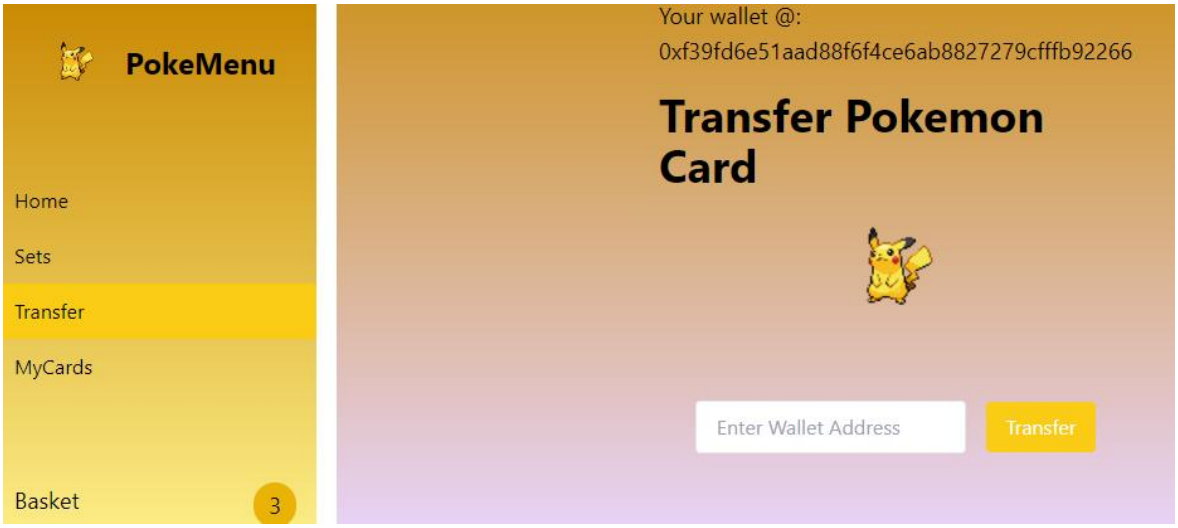

Figure 8 : Insertion de l'adresse de la Wallet recevant les 3 cartes à minter

En insérant l'adresse de la Wallet réceptrice, les cartes choisies seront envoyé à cette dernière.

Figure 9 : Mint de 3 cartes avec succès

Affichage des cartes mint dans notre collection « My Cards »

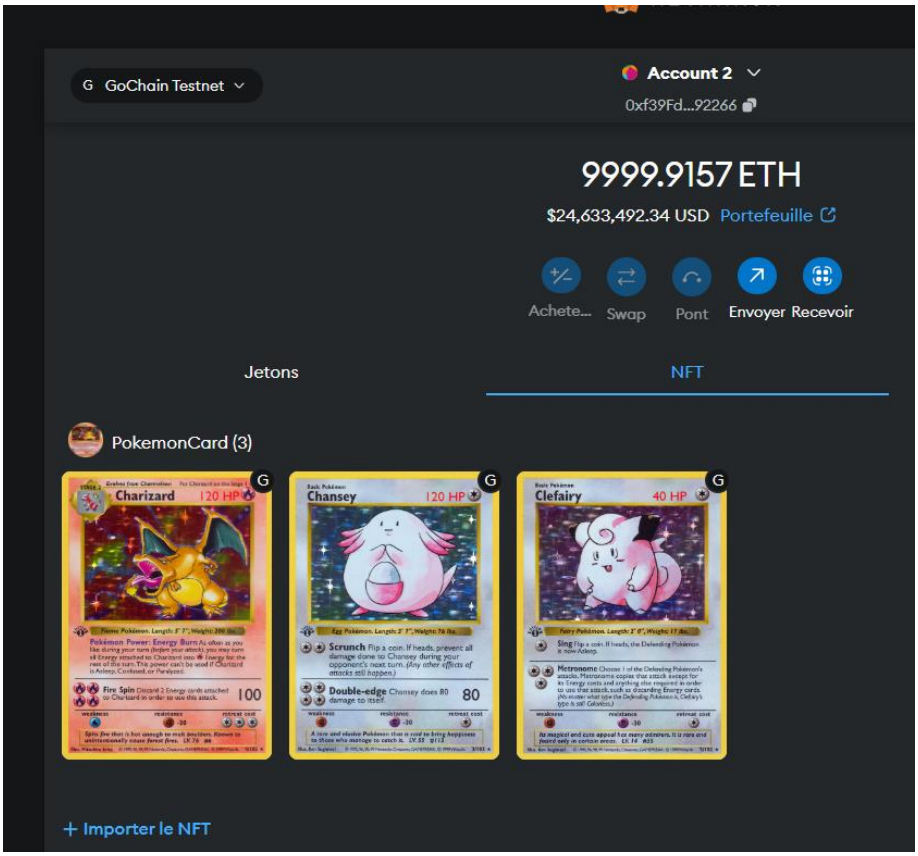

Figure 10 : Cartes uniques disponible dans notre Wallet

Les cartes choisies sont disponibles dans la wallet avec laquelle nous avons mint. Elles sont uniques et nous pouvons les vendre ou les donner à notre guise.

III. Perspectives :

- **Marketplace Décentralisé** : Créer un véritable marché interne où les utilisateurs peuvent acheter, vendre et échanger des cartes directement entre eux sans passer par un intermédiaire centralisé.
- **Création de Cartes** : Permettre aux joueurs de créer leurs propres cartes et de les mint sous forme de NFT, avec des frais de minting payés en ETH. Ces cartes pourraient être ajoutées à des collections ou échangées sur le marché.
- **Monétisation** : Mettre en place des achats intégrés permettant aux utilisateurs d'acquérir des packs de cartes, de booster leurs collections, ou de payer pour des fonctionnalités premium.

IV. Conclusion

Au cours de ce projet, nous avons acquis des compétences essentielles dans la création d'un jeu de cartes à collectionner décentraliser en exploitant les technologies blockchain. La mise en œuvre du standard ERC-721 pour les cartes en tant que NFT nous a permis de maîtriser la gestion des actifs numériques et d'assurer leur authenticité et leur traçabilité sur Ethereum. En parallèle, l'intégration d'API externes, comme celle de Pokémon TCG, a enrichi l'expérience utilisateur, nous donnant ainsi une compréhension approfondie de la conception d'architectures hybrides, reliant le on-chain et le off-chain de manière transparente. La construction d'un frontend dynamique et l'utilisation de Metamask pour gérer les connexions des utilisateurs nous ont permis de rendre les interactions blockchain accessibles et intuitives. Ce projet nous a donc permis d'approfondir nos compétences en développement blockchain et en architecture d'application, tout en renforçant notre expertise dans la gestion des NFT et des échanges décentralisés dans un environnement de collection.

Références :

[1] **Alina Novikova & Shuhan Duan**. *Collectible Card Game Project Repository*, GitHub. disponible : https://github.com/JerryProject/collectible-card-game-daar-inspi/tree/main.

[2] **CryptoZombies**. *Learn to Code Blockchain DApps by Building Simple Games*, Available at: https://cryptozombies.io/.

[3] **Solidity Documentation**. *Solidity 0.8.28 Documentation*, Available at: https://docs.soliditylang.org/en/v0.8.28/.

[4] **Pokémon TCG Developers**. *Pokémon TCG API Documentation*, Available at: https://pokemontcg.io/.