



Compte rendu projet 2 DAAR

Implémentation d'un jeu de cartes à collectionner sur Ethereum

Réalisé par:

Darko DJORDJEVIC

Melissa LATEB

Table des matières

1. Introduction.....	3
2. First on-chain part - Contracts.....	4
a. Création du contrat NFT:.....	4
i. Contrat Collection.....	4
ii. Contrat Main :.....	4
3. Second on-chain part - Backend.....	5
4. Off-chain development - Frontend.....	5
a. Page My Collections.....	6
b. Page Marketplace.....	6
5. Conclusion.....	8

1. Introduction

Le projet a pour but principal de créer un **jeu de cartes à collectionner décentralisé (CCG) sur la blockchain Ethereum**. Ce projet se concentre sur plusieurs aspects importants, notamment le développement des cartes de jeu sous forme de jetons non fongibles (NFT), respectant la norme ERC-721, ainsi que la création d'un marché où les joueurs peuvent échanger ces actifs numériques uniques. Un frontend convivial a aussi été conçu pour faciliter la gestion des collections de cartes, offrant une expérience utilisateur intuitive.

Le rapport explique en détail le projet, couvrant à la fois les aspects sur la blockchain (on-chain) et en dehors (off-chain). Il explore les choix de conception majeurs et décrit des étapes cruciales telles que la création des contrats NFT, la gestion des collections, l'attribution des cartes.

L'intégration de la technologie blockchain améliore l'expérience des jeux de cartes à collectionner en permettant des transactions sécurisées, transparentes et inviolables. Les cartes virtuelles deviennent des actifs uniques, échangeables et vérifiables. Chaque carte est représentée comme un NFT sur la blockchain, ce qui garantit son unicité. Elles peuvent être possédées, achetées, vendues sur des marchés basés sur la blockchain, assurant ainsi les droits de propriété des joueurs et des transactions sécurisées.

En résumé, le projet se concentre sur la création des cartes en tant que NFT et la mise en place d'un marché pour leur gestion, tout en posant les bases pour une gestion future des collections et des nouveaux ensembles de cartes.

2. First on-chain part - Contracts

a. Création du contrat NFT:

La première étape du projet consistait à créer un contrat NFT capable de gérer des collections de cartes. Chaque collection, ou ensemble, est composée d'un nom et d'un nombre de cartes, représentant un groupe de NFT. Nous avons utilisé la norme ERC-721 pour garantir que chaque carte soit un jeton non fongible, unique et identifiable sur la blockchain.

Nous avons également intégré l'API publique de **Yu-Gi-Oh!** pour récupérer les informations des sets et des cartes existantes dans cet univers, permettant ainsi de créer et de gérer automatiquement des collections thématiques basées sur des ensembles réels de cartes.

Chaque carte est associée à une collection spécifique. Le contrat NFT a été structuré pour permettre la création de ces collections via une fonction dédiée dans un contrat principal (*Main Contract*), qui centralise les informations relatives à toutes les collections créées.

i. Contrat Collection

Ce contrat gère la création et la gestion des cartes au sein d'une collection. Chaque carte est un NFT unique avec des informations telles que le nom, l'image, la rareté et l'état de rachat.

Fonctionnalités principales :

- **Mint de cartes** : Crée une nouvelle carte et l'associe à une adresse de propriétaire.
- **Récupération d'une carte** : Permet d'obtenir les détails d'une carte spécifique.
- **Mise à jour de l'état de rachat** : Permet au propriétaire de changer l'état de rachat d'une carte.
- **Transfert de cartes** : Gère le transfert de cartes d'un utilisateur à un autre, en vérifiant les permissions. (ventes et achats)

ii. Contrat Main :

Le contrat principal coordonne la création des collections et l'interaction avec les cartes via le contrat Collection. Il agit comme un gestionnaire centralisé pour toutes les collections et cartes du système.

Fonctionnalités principales :

- **Création de collections** : Permet à l'administrateur de créer une nouvelle collection. Les sets de cartes Yu-Gi-Oh! peuvent être intégrés automatiquement à partir de l'API pour former ces collections.

- **Mint de cartes à un utilisateur** : Permet de créer des cartes pour un utilisateur à partir d'une collection spécifique.
- **Récupération des informations sur une collection** : Permet d'obtenir des informations de base sur une collection (nom, adresse du contrat, nombre de cartes).
- **Récupération des métadonnées d'une carte** : Permet de récupérer les détails d'une carte spécifique au sein d'une collection.
- **Récupération des cartes détenues par un utilisateur** : Permet d'obtenir une liste de toutes les cartes détenues par un utilisateur à travers toutes les collections.

3. Second on-chain part - Backend

Structure et Fonctionnalité du Serveur server.js :

Le serveur backend est développé avec Express.js et expose plusieurs routes pour interagir avec les cartes de collection et le marketplace. Il initialise d'abord les collections et les cartes grâce aux fonctions **initCollections** et **initCardsToCollection**. Ces données sont ensuite accessibles via différentes routes.

- Initialisation des Collections : Lors du démarrage, le serveur appelle **initCollections** pour créer des collections, suivie par **initCardsToCollection** qui "mint" les cartes à différents utilisateurs.
- Endpoints Clés :
 - `/get-cards-by-address` : Récupère les cartes d'un utilisateur par son adresse.
 - `/get-all-collections` : Permet de récupérer toutes les collections existantes.
 - `/get-marketplace` : Affiche le marketplace avec les cartes disponibles à la l'achat.

Le serveur utilise également CORS pour permettre les requêtes externes et écoute sur le port 3000.

4. Off-chain development - Frontend

La structure de notre projet de jeu de cartes à collectionner est conçue pour répondre aux exigences du projet, offrant une expérience utilisateur centrée sur la gestion des collections de cartes tout en intégrant des fonctionnalités pour l'utilisateur et l'administrateur.

a. Page My Collections

La page **My Collections** permet à un utilisateur de voir toutes les collections de cartes qu'il possède. Chaque collection est affichée avec ses cartes correspondantes, et l'utilisateur peut vendre ses cartes pour qu'elles soient disponible dans le marketplace pour d'autres utilisateurs.

Comme dans la figure ci-dessous. (**Cette figure montrera la page My Collections avec les cartes d'une collection, et la mise en vente d'une carte, avec une confirmation de Metamask**)

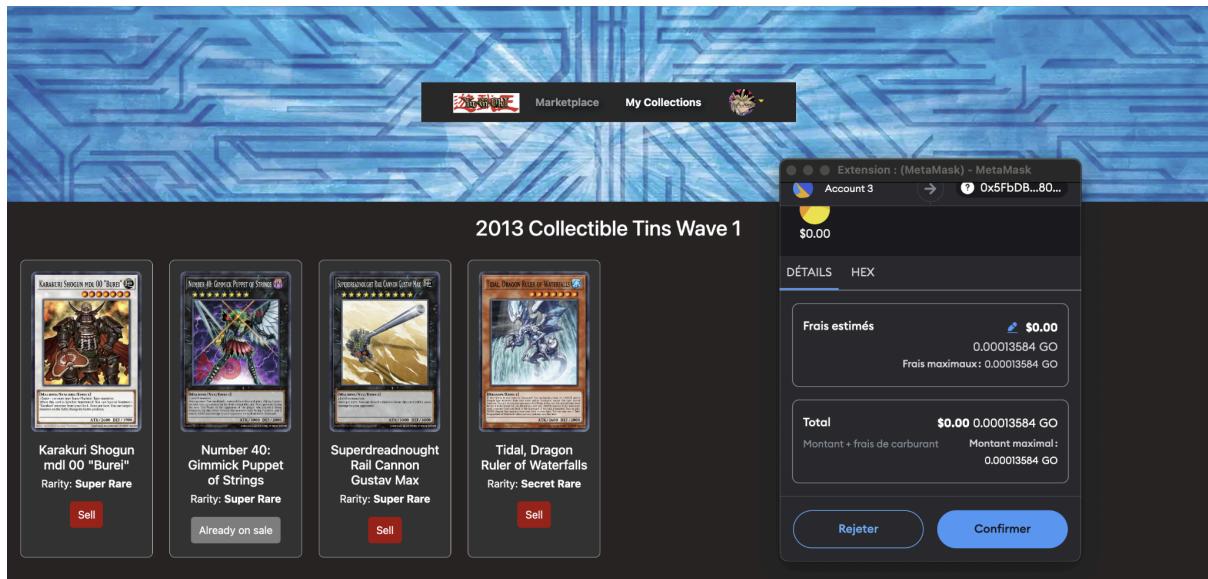


Figure 1 : Page My Collections affichant les cartes de l'utilisateur

b. Page Marketplace

Cette section permet aux utilisateurs d'accéder au marketplace où ils peuvent découvrir des cartes disponibles pour l'achat.

Dès qu'un utilisateur met en vente sa carte, celle-ci sera disponible à l'achat dans le marketplace, la carte sera disponible à l'achat auprès d'autres utilisateurs.

Comme dans les figures ci-dessous:

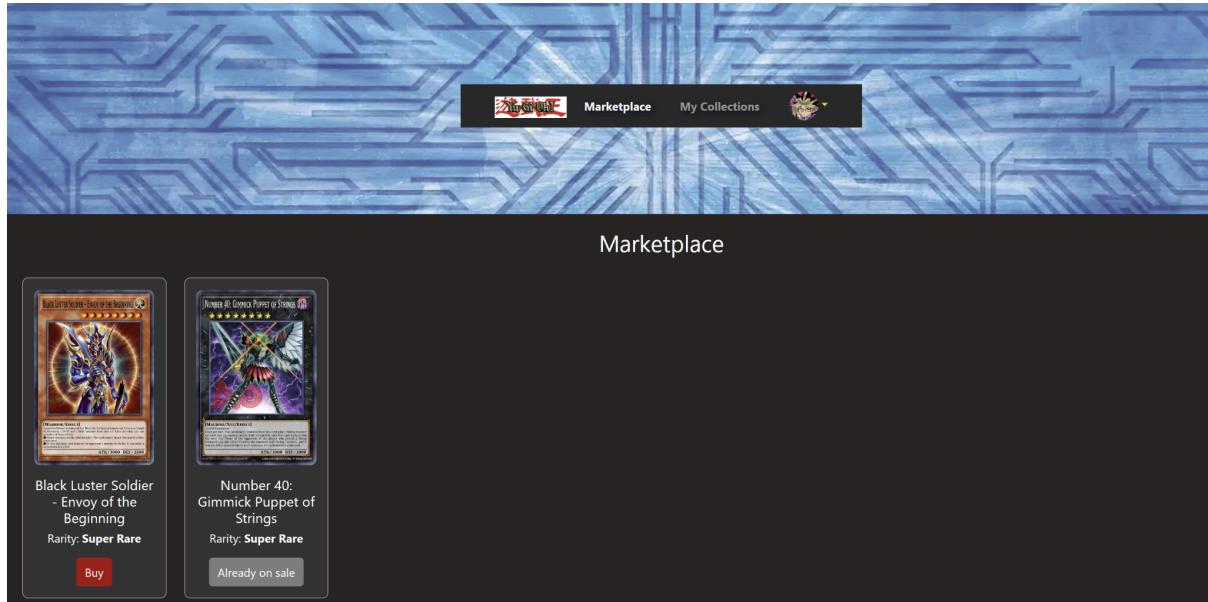


Figure 2 : Page Marketplace affichant les cartes que l'utilisateur a mis en vente / et les cartes mise en vente par d'autres utilisateurs

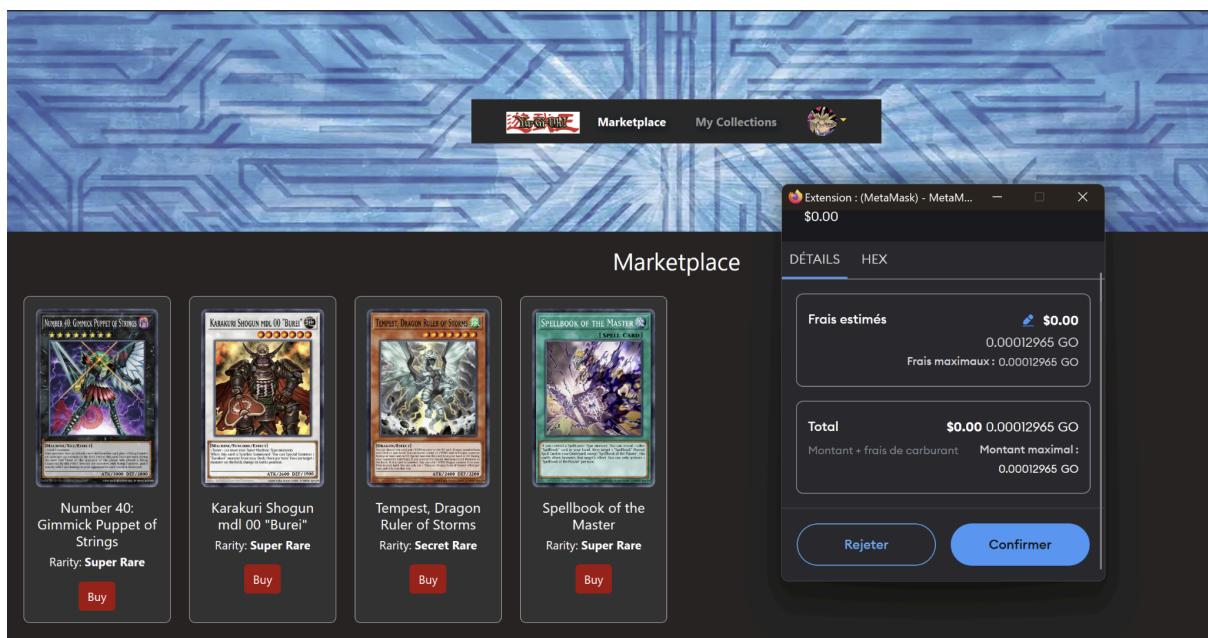


Figure 3 : Achat d'une carte par un autre utilisateur



Figure 4 : Collection de l'utilisateur après achat

5. Conclusion

En conclusion, ce projet nous a permis de créer une interface interactive dédiée aux passionnés de Yu-Gi-Oh TCG, en intégrant avec succès la collection, l'achat/vente et la visualisation de cartes. Cela a nécessité une maîtrise approfondie de React pour le front-end, ainsi qu'une intégration fluide et sécurisée avec les contrats intelligents afin d'optimiser l'expérience utilisateur.

Malgré les défis techniques rencontrés, nous avons su adopter des solutions innovantes, renforçant ainsi nos compétences en développement et affinant notre approche en programmation.

À l'avenir, nous souhaitons ajouter des fonctionnalités telles que les duels entre joueurs et l'échange de cartes, afin de reproduire fidèlement l'expérience Yu-Gi-Oh TCG. Ces améliorations promettent d'enrichir l'engagement des utilisateurs et de s'adapter à l'évolution du jeu de cartes Yu-Gi-Oh.