

UNIVERSITÉ SORBONNE



COMPTE RENDU DE PROJET DAAR

Collection de cartes NFT



Mohamed Amine ZEMALI et Thomas SANTONI
October 27, 2024

Contents

1	Introduction	2
2	Architecture	2
2.1	Main.sol	2
2.2	CardInstance.sol	3
2.3	Collection.sol	4
2.4	Résumé de l'architecture	4
3	Présentation de l'App	4
3.1	Page principale	4
3.2	Page de collections	6
3.3	MarketPlace	7
4	Conclusion	8

1 Introduction

Dans le cadre du cours DAAR (Développement des Algorithmes d'Application Réticulaire), ce projet vise à concevoir et à implémenter une application web de collection de cartes basée sur la blockchain Ethereum. Le projet se concentre sur la création de cartes à collectionner sous forme de NFTs (Non-Fungible Tokens), ce qui permet de garantir leur unicité et leur authenticité grâce à la technologie des contrats intelligents.

Les objectifs spécifiques de ce projet sont :

- Implémenter un système de création et de gestion de cartes NFT
- Développer un marketplace permettant l'échange sécurisé de cartes
- Mettre en place un système de collections et de boosters
- Créer une interface utilisateur intuitive pour l'interaction avec la blockchain

Les NFTs sont stockés sur la blockchain Ethereum, assurant ainsi la propriété numérique et l'échange décentralisé des cartes entre les utilisateurs. Ce choix technologique permet de répondre aux besoins d'une application de collection, où chaque carte représente un actif numérique unique et inaltérable.

Ce rapport présente les différents aspects de la réalisation du projet, depuis les choix technologiques jusqu'à l'implémentation du contrat intelligent et du front-end de l'application.

2 Architecture

L'architecture du projet repose sur trois contrats Solidity principaux : `Main.sol`, `CardInstance.sol`, et `Collection.sol`. Ces contrats interagissent pour gérer la création, l'attribution et l'échange des cartes de collection sous forme de NFTs (tokens non-fongibles) sur la blockchain Ethereum. Cette organisation permet de structurer les différentes fonctionnalités de manière modulaire et de faciliter la gestion des cartes et des collections.

2.1 Main.sol

Le contrat `Main.sol` est le point d'entrée principal de l'application et orchestre les différentes interactions entre les utilisateurs et les autres contrats. Ses principales fonctionnalités incluent :

- **openBooster**: Cette fonction permet aux utilisateurs d'ouvrir un booster et d'obtenir un nombre de cartes aléatoires à partir des cartes disponibles dans une collection spécifique. Pour chaque ouverture, un montant de 0.001 ether est requis, et deux cartes sont attribuées si suffisamment de cartes sont disponibles.

- **Gestion des collections:** `Main.sol` permet de gérer les collections de cartes, incluant l'ajout et l'initialisation de nouvelles collections, ainsi que la possibilité de récupérer les IDs et noms de chaque collection.
- **Interfaçage avec les autres contrats:** Ce contrat fait appel aux fonctions de `CardInstance.sol` et `Collection.sol` pour créer de nouvelles instances de cartes, assigner des cartes aux joueurs, et gérer les opérations au sein des collections.
- **Assignment des cartes:** Permet d'assigner une carte à un joueur. Une carte est créée en copiant les attributs de l'une des collections, puis assignée au joueur à l'aide de `CardInstance.sol`.
- **Ajout de carte:** Permet d'ajouter une carte au jeu en spécifiant sa collection, son nom, et son URL d'image. Chaque carte nouvellement créée est liée à une collection donnée.
- **buyCard:** Cette fonction permet d'acheter la carte d'un autre joueur, avec une vérification du prix, le transfert de fonds, et l'assignation de la propriété de la carte à l'acheteur.
- **Récupération des informations des cartes:** Les utilisateurs peuvent obtenir les détails des cartes qu'ils possèdent, comme le nom, l'ID, l'URL de l'image, le prix, et la disponibilité de chaque carte.
- **Gestion des prix et de la disponibilité:** Les propriétaires de cartes peuvent modifier le prix et la disponibilité de leurs cartes via les fonctions `changePrix` et `changeDispo`.
- **Retrait des fonds:** Le propriétaire du contrat peut retirer les fonds accumulés des ventes et ouvertures de boosters, en garantissant que les fonds sont sécurisés.

2.2 CardInstance.sol

Le contrat `CardInstance.sol` gère chaque carte individuelle en tant que NFT conforme au standard ERC-721, garantissant leur unicité et permettant leur transfert entre utilisateurs. Les fonctionnalités principales incluent :

- **Création de cartes:** Ce contrat définit les attributs de base des cartes, telles que leur identifiant unique, leur nom, et leurs caractéristiques spécifiques. Chaque carte est un NFT, et son propriétaire est stocké de manière sécurisée sur la blockchain.
- **Transfert de propriété:** En conformité avec le standard ERC-721, `CardInstance.sol` permet le transfert de propriété des cartes entre utilisateurs, assurant ainsi la sécurité et la traçabilité de chaque carte.
- **Intégration avec le contrat de collection:** Les cartes créées dans ce contrat peuvent être associées à des collections via le contrat `Collection.sol`.

2.3 Collection.sol

Le contrat `Collection.sol` est responsable de la gestion des collections de cartes, en définissant les cartes disponibles dans chaque collection et en assurant leur accessibilité pour les utilisateurs. Ses fonctionnalités principales sont :

- **getCurrentCardCount**: Retourne le nombre actuel de cartes dans la collection.
- **addCardToCollection**: Permet d'ajouter une carte spécifique à une collection, avec vérification que la collection n'a pas atteint sa capacité maximale.
- **getCard**: Permet de récupérer les informations d'une carte donnée dans la collection en fonction de son identifiant.

2.4 Résumé de l'architecture

Cette architecture modulaire permet de séparer clairement les responsabilités et d'assurer une meilleure maintenance du code. `Main.sol` sert de point de coordination pour l'interaction avec les utilisateurs, tandis que `CardInstance.sol` et `Collection.sol` gèrent respectivement les instances de cartes et leurs regroupements en collections. Ensemble, ces trois contrats offrent une plateforme complète pour gérer une collection de cartes NFT, de leur création à leur transfert entre utilisateurs.

3 Présentation de l'App

L'application a été faite en utilisant web3.js, nous avons lié le back et le front via des méthodes javascript. Dans le cadre du projet, nous avons utilisé l'API de Pokémon TCG (<https://dev.pokemontcg.io/>) pour récupérer les collections de cartes et les détails associés. Cette API fournit un accès à une base de données exhaustive des cartes Pokémon, ce qui nous a permis d'enrichir notre application avec des informations à jour sur les cartes, y compris leurs attributs, illustrations et autres caractéristiques. Malheureusement il y a énormément de carte, charger toute les cartes prendrait plus de 5 minutes à l'application pour se lancer, donc pour les tests nous ne lançons qu'avec 10 collections remplies.

3.1 Page principale

La page principale de l'application est le profil de l'utilisateur, il peut y retrouver les cartes qu'il possède. Dans cette page il peut chercher parmi ses cartes, les trier par nom/prix.

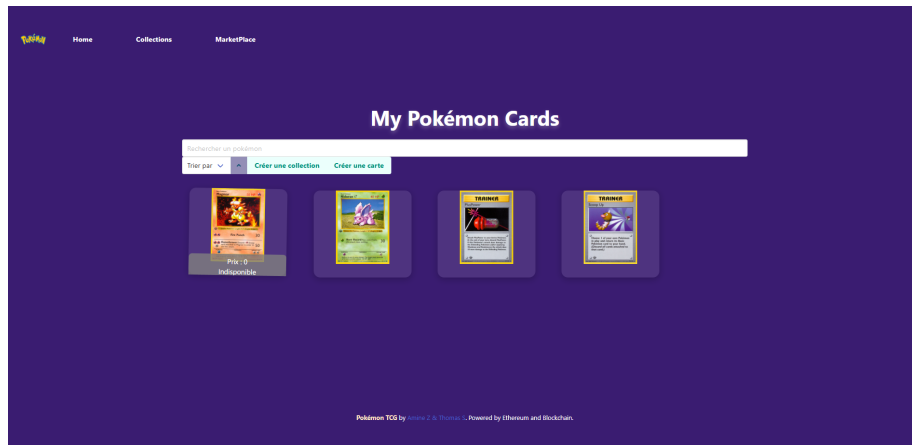


Figure 1: Page d'accueil de l'application

L'utilisateur peut également passer la souris sur une de ses cartes, la carte "bouge" légèrement et un petit bandeau apparaît lui indiquant le prix actuel de la carte et si il l'a mise à vendre sur le MarketPlace.

Si l'utilisateur clique sur une des cartes, il peut changer la mise en vente et le prix actuel de la carte.

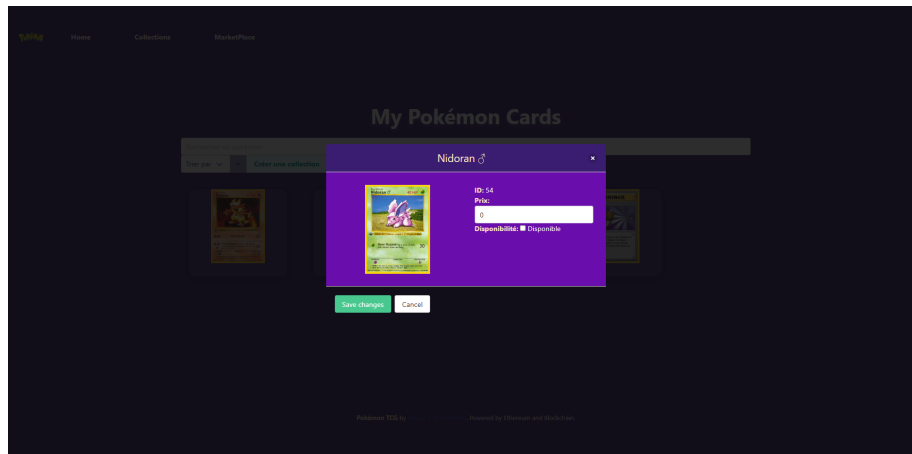


Figure 2: Fenêtre de modification de la carte

Sur la page d'accueil, on peut également ajouter une carte ou une collection, ce sont des boutons exclusifs à l'owner du contrat **Main.sol**.

Figure 3: Ajouter une collection

Figure 4: Ajouter une carte

3.2 Page de collections

La page de collection permet à l'utilisateur de visualiser toutes les collections et les cartes qu'elles contiennent. Le numéro de la collection est affichée ainsi que son nom. Au dessus de chaque carte, on peut y voir le logo de la collection, et en dessous le nom du Pokémon.

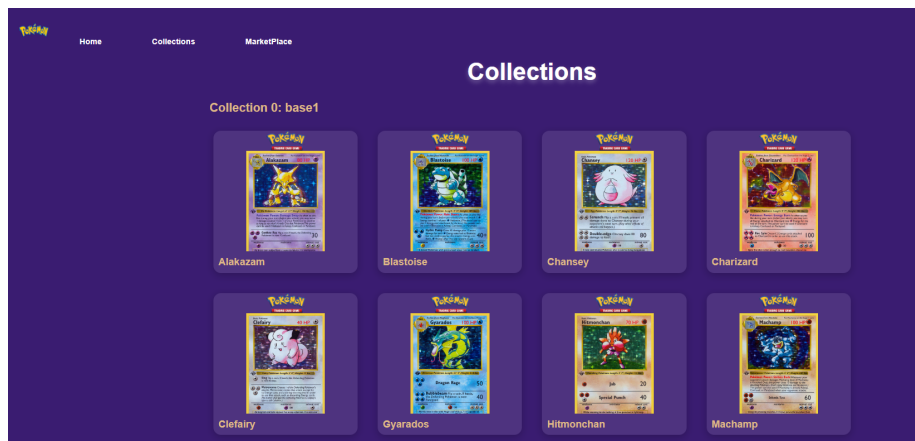


Figure 5: Page avec les collections

Si une carte plaît au joueur, il peut cliquer dessus et choisir combien de booster de la collection il veut acheter. Les frais d'achat sont à 5 ETH / booster, et les boosters contiennent une carte.



Figure 6: Fenêtre d'achat du booster

Le joueur pourra voir apparaître les cartes obtenues dans sa page d'accueil (et prier pour ne pas obtenir de carte énergie).

3.3 MarketPlace

La page la plus importante de notre application est le MarketPlace. Celui-ci affiche toutes les cartes disponibles à la vente, ainsi que leur prix. Les joueurs peuvent cliquer sur **Buy Now** pour acheter la carte. Se faisant, la carte est transférée de l'ancien utilisateur vers le nouveau, et sa disponibilité est mise en indisponible (pour ne pas qu'elle soit rachetée immédiatement derrière.)

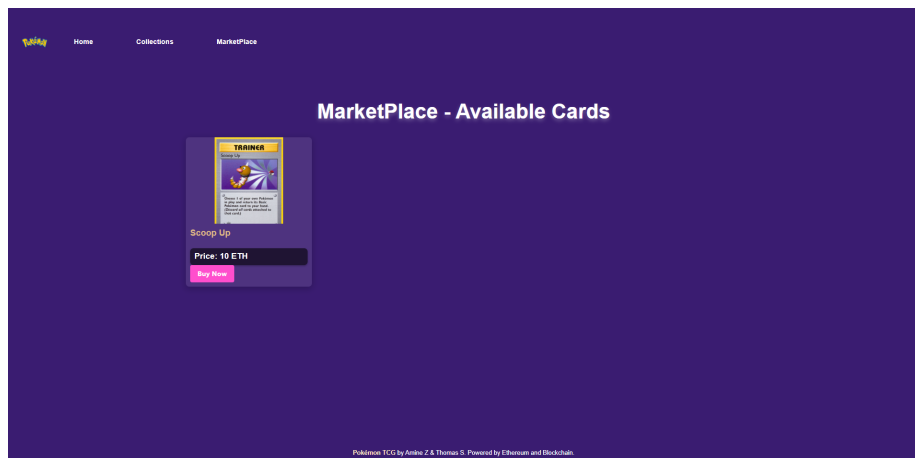


Figure 7: Page du marché des cartes

Pour assurer une meilleure expérience utilisateur, plusieurs fonctionnalités de sécurité ont été implémentées :

- Les propriétaires peuvent racheter leurs propres cartes pour corriger une mise en vente accidentelle par un utilisateur
- La carte devient automatiquement indisponible après l'achat pour éviter les doubles transactions et disparaît du MarketPlace
- Un système de vérification des fonds est effectué avant chaque transaction

4 Conclusion

Ce projet de collection de cartes NFT a permis de développer une plateforme complète combinant la technologie blockchain Ethereum avec une interface utilisateur intuitive. Les objectifs principaux ont été atteints avec la mise en place de trois contrats intelligents permettant la gestion des cartes NFT, un système de collections avec boosters, et un marketplace fonctionnel pour l'échange de cartes.

Cependant, nous avons rencontré des défis significatifs lors de la connexion du backend et du frontend, même après avoir suivi intégralement le tutoriel CryptoZombies. Cette difficulté a nécessité des efforts supplémentaires pour assurer une communication fluide entre les deux parties de l'application.

Les principales réalisations comprennent :

- L'implémentation réussie du standard ERC-721 pour la gestion des NFTs
- Le développement d'une interface utilisateur ergonomique
- La création d'un système de marketplace sécurisé
- L'intégration d'un mécanisme de boosters pour l'obtention de cartes