



27/10/2024

Master Science et Technologie Logiciel (STL)

Rapport de projet N° 2 : DAAR-Collective Card Game

Pokemon Card Game

Binôme :

Ghita Mikou 21423710

Zhengdao Yu 21304260

Enseignant :

Binh-Minh Bui-Xuan

Table des matières

Table des matières	2
1- Introduction	3
1-1 But de projet	3
2- Demo	3
2-1 Video de demo	3
2-2 Views	3
3- L'idée d'implémentation	7
3-1 Onchain part	7
Main contrat:	7
Collection contrat:	7
3-2 Offchain part	7
3-3 Integration of Pokémon TCG	7
3-4 Booster	8
3-5 To go beyond	8

1- Introduction

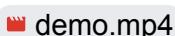
1-1 But de projet

Le but de ce projet est d'implémenter un jeu de cartes à collectionner de manière décentralisée sur Ethereum.

Le jeu est capable de créer les collections, les cartes(NFT), visualiser les collections et cartes correspondantes, mint carte en tant que admin, booster pack etc.

2- Demo

2-1 Video de demo



demo.mp4

https://drive.google.com/file/d/1C_re9Dv5pRINBOsXCsK5fGducNXS5qSO/view?usp=sharing

2-2 Views

View 1 - All collections pour le super-admin

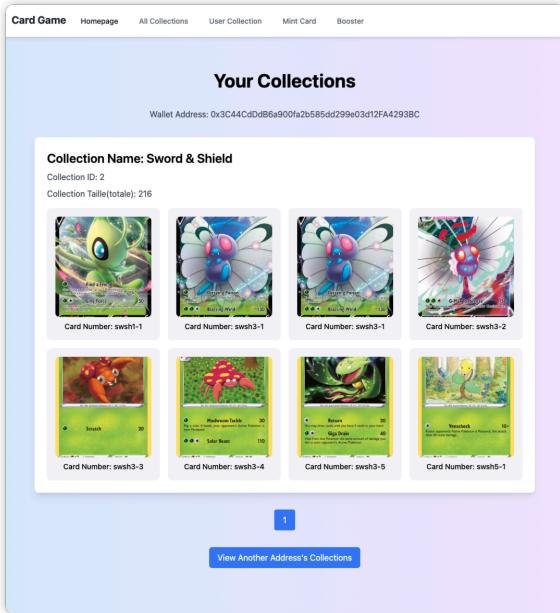
The screenshots show the 'All Collections' page of the Card Game application. The left screenshot displays the 'Sword & Shield' collection, which contains 12 cards. The right screenshot displays the 'CollectionVide' collection, which contains 5 cards. Each card is shown with its name, attack, defense, and type. The interface includes navigation buttons at the bottom of each page.

Fonctionnalités:

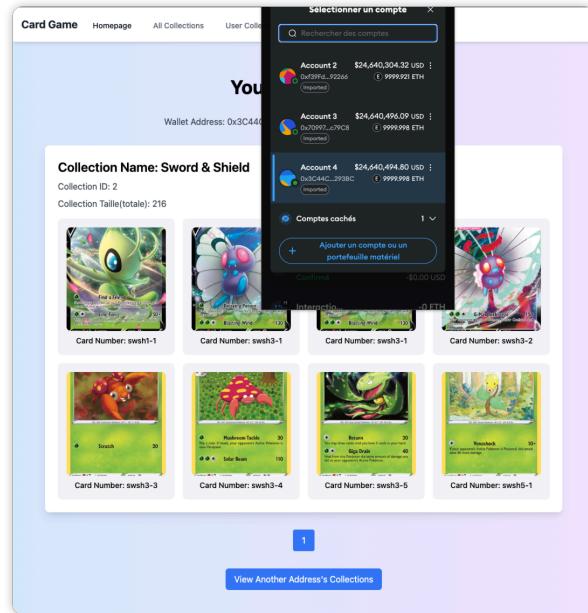
- Visualisez, pour chaque ensemble, tous les utilisateurs et leurs possessions
- Améliorez le frontend en ajoutant un moyen de gérer les collections, comme un classeur.
(On divise les collections page par page, switch boutons disponibles pour changer la page)
- L'affichage de : Collection Name, Wallet Adresse

View 2 - User collections

Après de connecter:

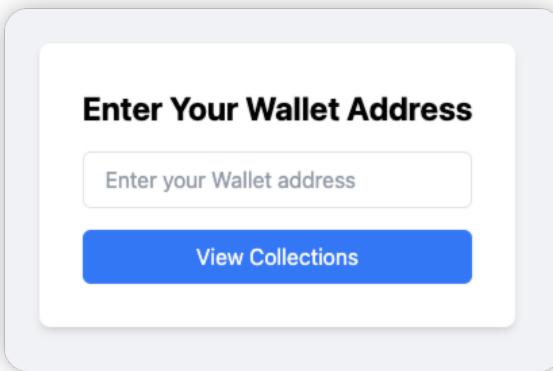


Connecter via metamask:



S'il y a des comptes connectés sur metamask, quand on clique sur "User Collection", la page détecte et récupère les infos de compte(Wallet) et se connecter automatiquement pour afficher ses propres cartes

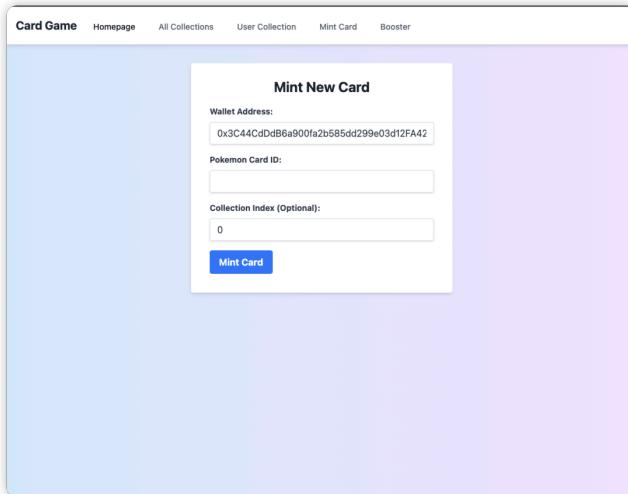
Si aucun compte connecté sur metamask, il faut taper l'adresse du Wallet manuellement:



Fonctionnalités:

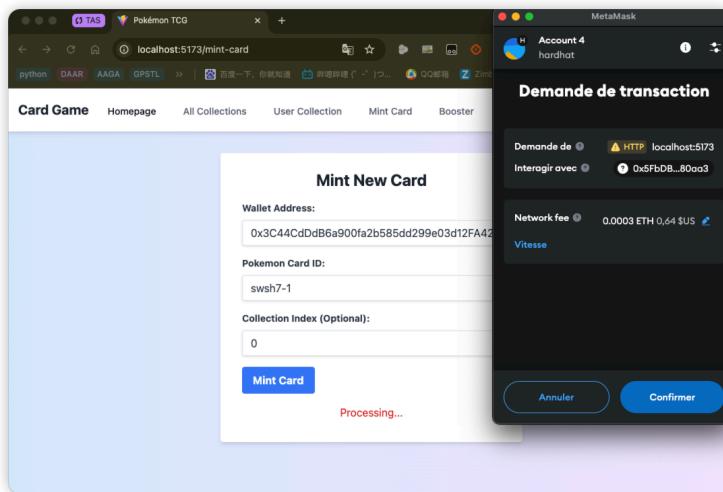
- Visualisez toutes les cartes (c'est-à-dire les NFT) possédées par l'utilisateur.
- L'utilisateur récupérera ensuite toutes les métadonnées des NFT à partir de l'API.
- Améliorez le frontend en ajoutant un moyen de gérer les collections, comme un classeur.
(On divise les collections page par page, switch boutons disponibles pour changer la page)
- L'affichage de :
 1. Collection Name, Collection ID (dans notre jeu, l'utilisateur peut avoir plusieurs collections), Wallet Adresse,
 2. MetaData of the NFT : Card image, Card Id (Card Number)

View 3 - Mint card dans la collection en tant que super-admin



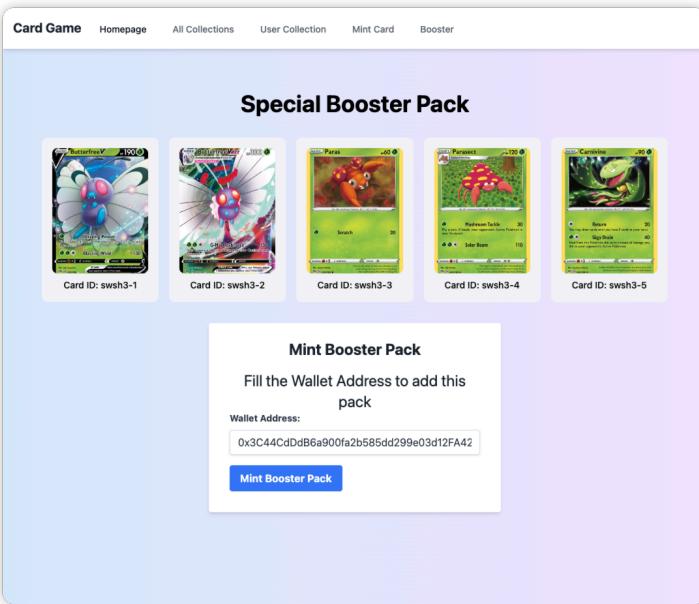
Fonctionnalités:

- L'affichage de : Wallet Address (remplie par défaut via metamask s'il y a des compotes connectés sur metamask)
- Taper le card id de Pokémon que on veut ajouter
- Choisir l'id de collection où il veut ajouter le card (en cas d'un utilisateur a plusieurs collections)



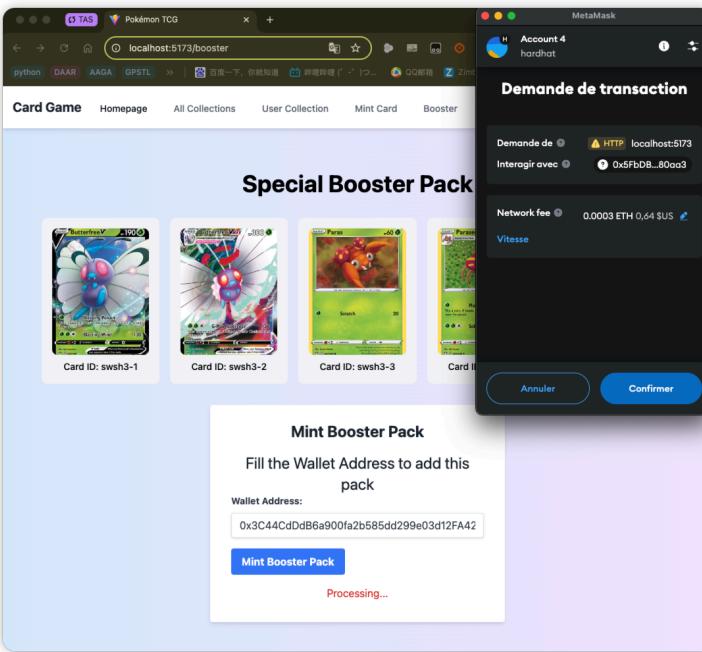
Dès que on veut mint une carte dans la collection,ça sera considéré comme une transaction dans metamask.Il suffit de cliquer sur confirmer ou annuler pour valider ou refuser cette opération.

View 4 - Booster Pack



Fonctionnalités:

- L'affichage de : Wallet Address (remplie par défaut via metamask s'il y a des comptes connectés sur metamask)
- Choisir le compte où il veut booster le pack de card (il va ajouter au premier collection d'utilisateur par défaut)



Comme le view mint-card ci-dessus, quand on clique sur "mint booster pack" pour échanger une pack de carte dans la collection, ça sera considéré comme des transactions dans metamask.

Chaque pack contient 5 cartes fixes, la page appellera donc 5 fois transaction dans metamask pour confirmer l'opération d'échanger les cartes.

3- L'idée d'implémentation

3-1 Onchain part

Main contrat:

fonctionnalités:

- créer collection
- mint carte
- getters (getCollection, getCollectionID...)
- récupérer les collections d'un user

Attention: On a décidé de ne mettre pas une page pour la création de la collection dans le frontend vu qu'il y a pas de grand intérêt et c'est pas imposé dans l'énoncé, on le laisse en backend (API disponible pour créer la collection)

Collection contrat:

fonctionnalités:

- mint carte
- récupérer tous les cartes de cette collection
- récupérer une carte par id de carte

3-2 Offchain part

On a réalisé 3 types de API dans notre projet:

1.Une API pour communiquer avec les contrats.

2.Une API pour interagir avec l'API de Pokémon afin de récupérer les ensembles (sets) et les cartes.

3.Une API qui combine les deux API ci-dessus pour initialiser l'environnement dès que le serveur démarre. (Par défaut : deux collections sont créées pour deux comptes différents ; l'une est remplie avec des cartes d'un set Pokémon, et l'autre reste vide pour jouer.)

3-3 Integration of Pokémon TCG

On a réussi à intégrer l'API de Pokémon TCG dans notre projet:<https://pokemontcg.io/>

3-4 Booster

Pour le booster pack, notre méthode consiste à considérer les boosters comme des **cartes spéciales (NFT)**.

Cela signifie qu'un booster sera considéré comme une carte spéciale avec des noms spécifiques (comme "booster1", "booster2") et sans image.

Nous allons stocker les "vraies" données contenues dans les boosters dans la partie Offchain (API). Pour échanger un booster, dès que l'API reçoit un booster, elle récupérera son nom, le recherchera dans le tableau des boosters et retirera les cartes correspondantes.

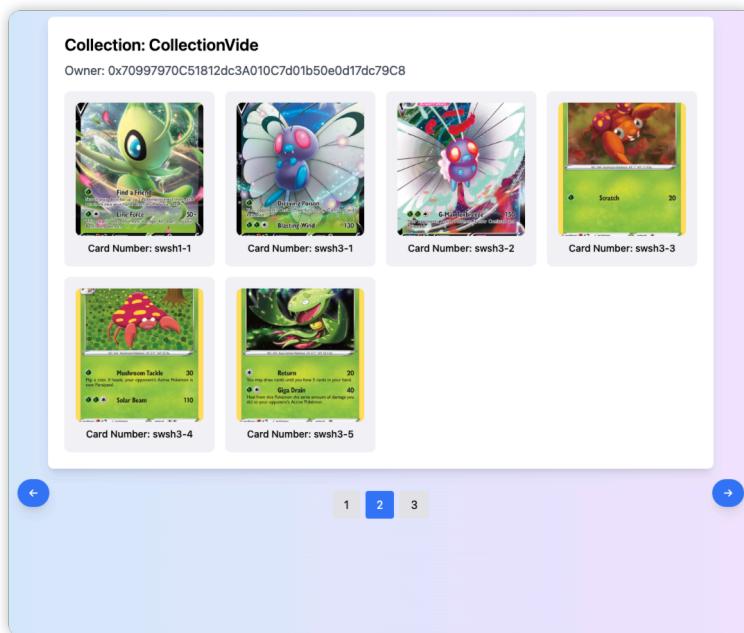
Après on peut simplement ajouter ces cartes de manière "mintcard".

3-5 To go beyond

On a réussi à implementer le premier feature :

Improve the frontend, by adding a way to manage collection, like a binder.

Comme les vues montrées ci-dessus:



On peut classer les collections page par page, et il dispose également des boutons pour changer la page.

