Cryptomon

Lifan Zhao

February 2020

1 Introduction

Cryptomon is a blockchain platform where players can lottery their cryptomons with Ethereum wallet, breed the new cryptomons from the ones they have owned, exchange cryptomons for the one they desire, and fight against each other's cryptomon.

Therefore, the platform is split into four different page: Trade, Breed, Fight, Profile:

- **Trade**: Trade page is designed for gaining cryptomons. The ways includes lottery, buy cryptomons and share cryptomons.
  - Lottery: By paying **lotteryFee** amount of WEI to the owner of the contract, the players can gain randomly five cryptomons from the contract's. **baseCryptomons**
  - Buy: The cryptomons that other players want to sell will be listed here. One can buy these cryptomons by paying the tagged **price** to the provider player.
  - Share: Since the breed mechanism and lottery, players may need some cryptomons with specific requirements and at the mean time get some duplicated cryptomons. They can share the one that no longer needs and raise the exchange requirement. Hence, all the shared cryptomons are listed here, only those who can provide the required cryptomon can completed the exchange. It will not charge any fee, other than gas.
- **Breed**: Breed page is designed for giving birth of new cryptomons. The breeding mechanism is designed in the following:
  - Only cryptomons belongs to the same **pokedexId** and from two different genders can breed.
  - The new cryptomons belongs to the same **pokedexId** as its parent. Its level will be the half of lower **level** of its parents. And the points from each layer will be distributed randomly to its **HP**, **ATK** and **DEF** attribute.
  - After successfully breeding, both of its parent require time to cool down.
- **Fight**: Fight page is designed for cryptomons fighting and level up. One can set his cryptomons as **forFight** so that they can be listed here for other players to attack. As a reward for being chosen, these cryptomons is the first to attack during fighting.
  Unfortunately, for the security reason that miner can view and utilize the random function to win every fight, in Cryptomon every fight is predictable. The fighting mechansim is designed in the following:
  - The listed cryptomons attack first. The damage is attacker's **ATK** subtracted by victim's DEF. If DEF is larger than **ATK**, then no damage applies.
  - Two cryptomons will fight until one's HP become zero.
  - The winner will slowly recover its **HP** to **maxHP**, while the loser will have to be remain unavailable until **requiredCoolDown** time elapse.
  - The winner will gain one **winPoint**. And with 10 **winPoint**, the cryptomons is able to level up. When levelup, the cryptomons can add one point to the attribute the play wants. Yet, at the same time, its **requiredCoolDown** will be extended.
- **Profile**: In Profile Page, players can see all of cryptomons they owned stated with different status. Players can also set price to **sellCryptomons**, set requirement to **shareCryptomon** or **setForFight**. Cancel options is also provided here.

Waht's more, there is a set of **baseCryptomons** on blockchain to which only owner of the contract can added new cryptomons. **baseCryptomons** is private to all. Owner only can only perform add action which I think can be good way to store so many pokemons without exceeding the size limit of smart contract.

2.1 Solidity

The platform is built on Solidity. Again, Cryptomons Contract is splited into 4 different part:

- cryptomons.sol: in charge of cryptomons storage and creation function.
  - **buildCard(uint32 pokedexId, uint32 HP, uint32 ATK, uint32 DEF, uint32 requireCoolDown) public onlyOwner** which can only be called by owner to build the baseCryptomons where normal cryptomons get their settings from.
  - **createCryptomon(uint pokedexId, uint16 level) internal** which can only be called from other two public functions "breedCryptomons" and "lotteryCryptomons".
  - **lotteryCryptomons** which transfer amount of WEI to the owner of contract and get 5 random cryptomons from baseCryptomons with random gender and level 1.
  - some helper function to get mapping in struct.
  - **rand(uint256 _length, uint seed)** which use block.difficulty, block.timestamp and an input seed as random seed and returns random number from required range(length). Indeed, it is not safe. Yet, nothing important related to this random function, therefore I use it directly to decide gender and pokedexId.
- cryptomonsBreed.sol: in charge of cryptomons breed and related functions.
  - **breed(uint cryptomonId1, uint cryptomonId2) public onlyOwnerOf(cryptomonId1) onlyOwnerOf(cryptomonId2)** which requires msg.sender as the owner of two selected cryptomons, check the breeding requirements, generated a new cryptomon and set two cryptomons to cool down.
  - **addHP/addATK/addDEF(Cryptomon storage cryptomon, uint16 pointsToAdd) internal** which add level up points to attributes.
- cryptomonsFight.sol: in charge of cryptomons fight and related functions.
  - **fightCryptomon(uint target, uint cryptomonId) public onlyOwnerOf(cryptomonId) readyFighting(target, cryptomonId)** which requires cryptomons[cryptomonId] belongs to msg.sender and cryptomons[target] is either not for sale, for share, or cooling down. And calculate the fight result, give the winPoint to winner and set loser to cool down.
  - **setForFight(uint id, bool forFight) public onlyOwnerOf(id)** which only allow the owner of cryptomon to change the fight status.
- cryptomonsTrade.sol: in charge of cryptomons sell, purchase and exchange functions.
  - **sellCryptomon(uint id, uint price) public onlyOwnerOf(id)** which only allow the owner of the cryptomons to set the price and put it on market.
  - **retreiveCryptomon(uint id) public onlyOwnerOf(id)** which only allow the owner of the cryptomons to cancel the sale and remove it from market.
  - **buyCryptomon(uint id) public payable** which allow players to buy the cryptomons with its tagged price.
  - **sharingCryptomon(uint id, uint32 requiredPokedexId, uint32 level, uint32 gender) public onlyOwnerOf(id)** which only allow owner of the cryptomons to share it with stated requirements and put it on market.
  - **exchangeCryptomons(uint id, uint target) public onlyOwnerOf(id)** which allow the player to exchange the shared Cryptomon on market with the satisfied cryptomons he have.
  - **cancelSharingCryptomon(uint id) public onlyOwnerOf(id) returns (bool)** which remove the cryptomons from sharing market and reset its exchange requirements.

2.2 Web Interface

The frontend is implemented with React and web3 library. No backend is implemented, because web3's interactions with Cryptomon Contract give the enough functionalities.