CryptoCampo Project

Smart Contract Documentation

Author: Eduardo Mannarino
Update: 11/09/2022

General Considerations

A single contract will be developed that will be responsible for the NFTs and all the necessary on-chain tasks (will be detailed later).
OpenZeppelin libraries will be used (https://openzeppelin.com/).

- **ERC721:** standard for Non Fungible Token (NFT).
- **ERC721Enumerable:** Extension to be able to list tokens by owner.
- **Ownable:** Module for access control to administrator functions.
- **ReentrancyGuard:** Module to avoid reentrancy in functions.

Variables

baseURI: Stores the URI for Metadata (if needed).
buyers: Mapping to store buyers and prevent a second buy.
buyFee: Commission percentage for token purchase. Uses 2 decimal places (1% = 100).
canBuy: Indicates whether tokens can be generated.
canClaim: Indicates whether tokens can be claimed.
canReturn: Indicates whether tokens can be returned.
canReBuy: Indicates whether tokens can be rebuyed.
feesCollector: Address (wallet) to indicate the fees collector.
fundsToken: Address to indicate the currency of collection and payment. It has to be ERC20.
fundsCollector: Address (wallet) to indicate the funds collector.
MAX_SUPPLY: Maximum amount of NFT to be minted.
NFT_VALUE: Value of each NFT.
profitToPay: percentage of profit to be returned. Uses 2 decimal places (20% = 2000).
reBuyFee: Percentage of charge for rebuying a NFT.
returnCollector: Address (wallet) to indicate the return collector.

returnFee: Percentage of discount for returning a NFT.
tokenBurned: Amount of NFT burned.
tokenCount: Amount of NFT minted.

Functions

Buy

Description
Allows the user to invest in the platform by generating the corresponding NFT.

Parameters
No parameters.

Requirements
User has approved the contract to use their payment token
canBuy is true.
FundsCollector has been set.
FeesCollector has been set.
TokenCount must be less than MAX_SUPPLY.
Address have not bought before.

Action

TokenCount is incremented by 1.
Register the address as a buyer.
Mint the NFT to the transaction sender.
The value of the NFT is transferred from the user to the Collector Wallet.
The value of the Buy Fee is transferred from user to Fees Wallet
The Buy event is emitted.

Mint

Description
Allows the owner of the contract to mint NFTs for free and send them to a specific address.

Parameters
address: address to send the NFTs.
amount: Amount of NFT to mint.

Requirements
Address is not Zero Address.
TokenCount must be less than MAX_SUPPLY.

Action
Mint the amount of the NFT through a loop.

ReturnToken

Description
Allows the user to return the NFT to CryptoCampo.

Parameters
tokenId: ID of the token to return

Requirements
canReturn is true.
FundsCollector has been set.
ReturnCollector has been set.
The token ID exists. (Generated and not burned).
User is the owner of the NFT.

Action
The token is transferred from user to ReturnCollector.
The value of the NFT minus the return fee is transferred from Collector Wallet to User Wallet.

ReBuy

Description
Allows the user to buy a token returned.

Parameters
tokenId: ID of the token to buy.

Requirements
canReBuy is true.
The token exists. (Generated and not burned).
The NFT is owned by Return Wallet.

Action
The token is transferred from ReturnCollector to User.
The value of the NFT plus the rebuy fee is transferred from User Wallet to Collector Wallet .

Claim

Description
Allows to claim several tokens, recovering their value plus the corresponding profit.

Parameters
listTokenId: list of token IDs to claim.

Requirements
canClaim must be true.
That the tokens exist. (Generated and not burned).
That all tokens are owned by the claimant.

Action
All indicated tokens are burned.
The TokenBurned is incremented by the amount of NFT to be claimed.
It is transferred from the Collector Wallet to the User Wallet for the total of the funds plus the profit to be paid.
The Claim event is emitted.