CryptoCampo Project

Smart Contract Documentation

Author: Eduardo Mannarino
Update: 10/05/2022

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 forNon 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.
- **Counters:** Utility for counters.

Variables

buyFee: Commission percentage for token purchase. Uses 2 decimal places (5% = 500).
canBuy: Indicates whether tokens can be generated.
canClaim: Indicates whether tokens can be claimed.
canTrade: Indicates whether tokens can be exchanged.
fundsToken: Address to indicate the currency of collection and payment. It has to be ERC20.
fundsCollector: Address (wallet) to indicate the funds collector.
feesCollector: Address (wallet) to indicate the fees collector.
listTokensOnSale: Array that stores the IDs of tokens for sale, for sequential access.
maxBatchCount: Maximum amount of tokens to buy or claim per batch. This limit is intended to avoid exceeding the maximum amount of gas.
maxValueToRaise: Maximum value of total tokens. (Maximum amount of investment to receive). Uses 18 decimal places (1 USD = 10 ^18).
profitToPay: percentage of profit to be returned. Uses 2 decimal places (20% = 2000).

tokenIdTracker: Sequential ID handler for token identification.
tokensOnSale: Mapping to store the tokens for sale and their eventual price.
totalValue: Indicates the total value of all tokens. Uses 18 decimal places (1 USD = 10 ^18).
tradeFee: Percentage of commission for token exchange. Uses 2 decimal places (1% = 100).
validValues: Array to indicate the valid values of tokens. (Ex: 100, 500, 1000, etc). Uses 18 decimal places (1 USD = 10 ^18).
values: Mapping to store the value of each token.

Functions

Buy

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

Parameters
value: Token value (unique value for each token). Uses 18 decimal places (1 USD = 10 ^18).
amount: Amount of tokens to buy.

Requirements
User has approved the contract to use their payment token
canBuy must be true.
The quantity to be purchased is less than the maximum quantity per batch.
The value is one of the valid values.
The total to buy does not exceed the total value of tokens allowed.

Action

The total value of tokens is added to totalValue.
The total to be acquired is transferred from the buyer's wallet to the funds collector wallet. (Quantity * Value).
The value corresponding to the commission is transferred from the buyer's wallet to the fees collector wallet (Total Value * buyFee / 100)
The corresponding NFTs are generated and transferred to the buyer's wallet.
The Buy event is emitted.

PutOnSale

Description

Allows the user to put his token up for sale at a certain price.

Parameters

tokenId: ID of the token to put up for sale.
price: price fixed for the sale. Uses 18 decimal places (1 USD = 10 ^18).

Requirements

canTrade must be true.
The token ID exists. (Generated and not burned).
The token to be put up for sale is the property of the user executing the function.

Action

The token is registered as available for sale and its corresponding price.
The PutOnSale event is emitted.

RemoveFromSale

Description

Allows the user to remove their token from the sale.

Parameters

tokenId: ID of the token to remove from sale.

Requirements

canTrade must be true.
The token ID exists. (Generated and not burned).
The token to be removed from the sale is the property of the user executing the function.

Action

The token is removed as available for sale.
The RemoveFromSale event is emitted.

Trade

Description

Allows to buy a token on sale.

Parameters

tokenId: ID of the token to buy.

Requirements

The buyer has approved the contract to use their payment token.
canTrade must be true.
The token exists. (Generated and not burned).
The buyer is not the seller.
The token is on sale.

Action

The established price is transferred from the buyer's wallet to the seller's wallet.
The commission amount (Nominal value * tradeFee / 100) is transferred from the buyer's wallet to the fee collector wallet.
The token is transferred from the seller to the buyer.
The token is removed as available for sale.
The Trade event is emitted.

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.
The amount to be claimed is less than the maximum amount per batch.
That the tokens exist. (Generated and not burned).
That all tokens are owned by the claimant.

Action

The total value of tokens is subtracted from totalValue.
All indicated tokens are burned.
It is transferred from the fund collection wallet to the claimant's wallet for the total of the funds plus the profit to be paid.
The Claim event is emitted.