OctoPlace v1

NFT Marketplace

Jared Borders

December 26, 2021

Abstract

This technical whitepaper explains some of the designs decisions behind OctoPlace v1 contracts. It covers basic features and trade-offs made due to time restrictions. This whitepaper also depicts future improvements and ideal features which are beyond the scope of this current version.

1. Introduction

OctoPlace v1 is an on-chain system of smart contracts on the Ethereum blockchain allowing for the buying, minting, and selling of Non-Fungible Tokens (NFT) defined by both ERC721 and ERC1155 token standards. The three contracts which define this functionality are the following: Market.sol, ERC721Tradable.sol, ERC1155Tradable.sol. OpenZeppelin's open-source contracts were used for ERC721 and ERC1155 token implementations which were used as needed in the above contracts. OctoPlace v1 specifically aims to support a user minting a new NFT and listing this token for a set price on-chain which can later be bought. Both bought and created NFTs can be viewed by a user whose browser wallet is connected to the decentralized application (dApp). By uploading images to the Interplanetary File System (IPFS), the image location (uniform resource identifier (URI)) can be stored on-chain and point to an asset existing off-chain. As previously mentioned, users have the option to mint an NFT which follows either the ERC721 or ERC115 standards. The only difference *seen* from the user's point-of-view is the different tabs in the UI which were purposely separated to ensure better user awareness and experience. The contract and front-end architecture were influenced by the following systems/protocols/documentation: OpenSea's developer tutorials, OpenZeppelin's documentation, and Nader Dabits' article(s) on dApp development. Finally, OctoPlace v1 is currently only deployed to Kovan test network.

2. Instructions

Connecting your wallet to the dApp will allow you to mint new tokens, view NFTs you created, and view tokens you own. A user can see NFTs currently in the market (i.e. tokens which have not been sold). Connecting your wallet also allows you to buy NFTs listed in the market for the price associated with them. Finally, once a user purchases an NFT, that NFT is no longer listed in the market, therefore, preventing others from being able to buy it.

3. Future

OctoPlace v1 supports buying NFTs which follow either ERC721 or ERC1155 token standards, but some of the features which separate the two are not exposed in this dApp. A limited example: ERC1155 supports batch transfers and fungible token minting but OctoPlace v1 only supports single token transfers/minting. Given the purpose of this project and the time restrictions, developing features of that complexity were not treated as a top priority. However, the foundations are present to easily allow for future iterations to include this functionality.

Future implementations would allow a user to re-list an NFT they own. Several other quality-of-life upgrades exist, such as increased differentiation between the token standards and increased customization when a user wishes to mint a new NFT.

OctoPlace v1 front-end has not been thoroughly tested and serves as a basic means of interacting with the contracts defining the NFT marketplace. Edge-cases relating to listing price, image size, image format, etc. may in fact result in unexpected behavior. As stressed above, a full-scale, fully-tested application is beyond the scope of this project, but an option in the future. Finally, metamask was the only wallet used for development and testing. Other wallets may result in unexpected behavior.

4. Testing

The contracts defining OctoPlace v1 functionality were thoroughly tested. Market operations such as minting tokens, buying/selling tokens, and retrieving token data are covered via the integration testing found here: https://github.com/JaredBorders/OctoPlace/blob/master/test/integration/market.behavior.js.

References

IPFS: https://ipfs.io

OpenSea: https://docs.opensea.io/docs

OpenZeppelin: https://openzeppelin.com/contracts/

Other (Repositories):

1. https://github.com/dabit3
2. https://github.com/ProjectOpenSea/opensea-creatures
3. https://github.com/ProjectOpenSea/opensea-erc1155

Uniswap v3 whitepaper: https://uniswap.org/whitepaper-v3.pdf

OctoPlace v1 Contract URLs:

Market.sol: https://github.com/JaredBorders/OctoPlace/blob/master/contracts/Market.sol

ERC1155Tradable.sol: https://github.com/JaredBorders/OctoPlace/blob/master/contracts/ERC1155Tradable.sol

ERC721Tradable.sol: https://github.com/JaredBorders/OctoPlace/blob/master/contracts/ERC721Tradable.sol

Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. This paper reflects current opinions of the author. The opinions reflected herein are subject to change without being updated.