CS 646 Group 2 DApp With Ethereum Smart Contract

King, Tiara   Nettles, AJ   Allison, Leigh
@uab.edu   @uab.edu   leighg@uab.edu

November 29, 2022

Abstract

We introduce Blaze Sneakers, a decentralized web application (DApp) for purchasing sneaker images as non-fungible tokens (NFTs). The Blaze Sneakers web site allows users to buy images of popular sneakers as tokens (SNKRs) on an Ethereum network. Once an account has purchased at least one SNKR, the account owner can view their entire collection on the site. NFTs are increasing in popularity, and the proof of ownership provided by blockchain technology enables many uses. They could be used to grant access to special web sites or virtual events. The number of tokens available is initially limited, but we can add new images/tokens at any time.

I. Introduction

II. Background

Blockchain technology was first introduced with the paper "Bitcoin: A Peer-to-Peer Electronic Cash System", published by Satoshi Nakamoto in 2008. Satoshi Nakamoto is a fake name, however, so we do not know for certain who invented it. Blockchain is a data storage structure that is similar to a linked list, but it is impermeable. Each "block" is made up of transactions and is built off the previous block, creating a chain that cannot be altered. It enables the exchange of virtual currency that does not require a trusted third party. It operates in a decentralized fashion, on all computers that are connected its network, as opposed to on a centralized server. Ethereum is one implementation of blockchain and cryptocurrency.

III. Project Description

The front end of the project is a user interface that enables clients to browse and purchase NFTs. The back end is a smart contract, deployed on an Ethereum

blockchain. When a user visits the home page (*Figure* 1), it displays a list of all NFTs currently in the system. Each one is labeled as having been sold, available for purchase, or coming in the future. This availability of each token is obtained by calling a function in the contract.
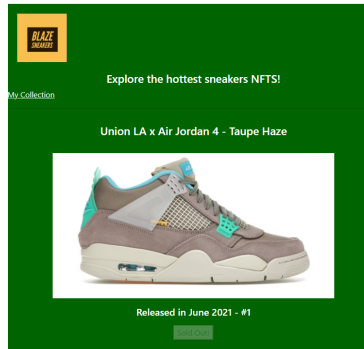

Figure 1: Blaze Sneakers Home Page

If an NFT is available for purchase, a button is enabled to allow the user to purchase it. When the purchase button is clicked, it will prompt them with the price and asks if they are certain. When a user confirms their intention to purchase a SNKR, another function in the smart contract is called to mint it. This function handles the transfer of funds from the user's account to the contract owner's account. The user is directed to log in to their Ethereum wallet and choose the account with which to pay (*Figure* 2).
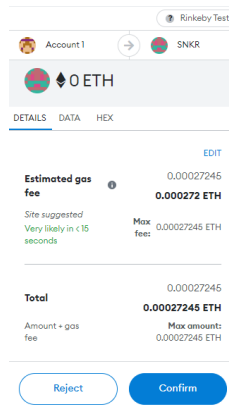

Figure 2: MetaMask Purchasing Example

Once they own at least one SNKR, they can view their collection via the "My Collection" link. This link goes to a separate page for displaying all SNKRs owned by an account. It uses a verify function in the smart contract to determine

which tokens are owned by the account.

To add new tokens to the site, we visit a hidden control panel page. This page calls a contract function to confirm whether the user is the contract owner. If they are, a button will be displayed for incrementing the number of tokens. This button calls another function in the contract to increase the total number of SNKR tokens.

IV. Design Decisions

We chose to write the Ethereum smart contract in the Solidity language and used the Remix online integrated development environment. The Solidity code must be compiled into bytecode that runs on the Ethereum network. Remix allows us to compile, test, and deploy the contract in one environment. We deployed the contract on the Rinkeby testnet. This allows for risk-free testing on a network that does not user real ether.

Using the ERC721 standard for NFTs makes the contract interoperable with others who follow the standard. OpenZeppelin has a template for ERC721, which makes developing basic NFT functions and following the standard less complex.

The NFT image files are stored on the Interplanetary File System (IPFS), as well as in the project's repository. IPFS is a decentralized way to store the images, but our implementation requires the user to have IPFS installed and running locally.

For the front end application, we used HTML, CSS, and Javascript to customize the pages. This was done using Visual Studio Code as a development environment. The web3.js package is provided to handle the connection to the Ethereum network. In order to host the site for testing, we used the LiveServer plugin by Ritwick Dey.

V. Limitations and Future Work

IPFS must be installed and running on the client device for the images to be available via IPFS. In the future, we would like to make the contract itself able to access the images. This would alleviate the requirement of IPFS on the client. Future work: NFT ownership could allow access to private web sites and online events.

VI. Conclusions

Blaze Sneakers is the place to go for sneaker NFTS.

VII. References

Antonopoulos, Andreas M. and Dr.Gavin Wood. 2019. Mastering Ethereum, https://github.com/ethereumbook/ethereumbook. (2022).