2023-02-01

NFT AutoBidder

Kristoffer Wallqvist

Degree Project BCU22D
Medieinstitutet

Table of contents

Project idea summary

I came up with this idea to scratch my own itch as an NFT collector and trader. I have built a bot that will place automatic WETH offers on NFT collections based on parameters entered in the frontend. The parameters are collection slug (in other words, which NFT collection), a max bid entered by the user, and a minimum difference to the floor price entered by the user. So say the floor is 1 eth and the user says they want to bid max 0.8 eth, and the current top bid is 0.74, then the bot will bid 0.7401 eth. It places bids that are valid for 15 minutes. When those 15 minutes are up, the bot fetches floor price and top bid again, and places a bid if the parameters are right. This is a great way to get discounted NFTs for users who are not in a rush and are not too concerned about which NFT in the collection they get.

Alterations from the project plan

There has been an alteration from the project plan. Instead of the user depositing into a smart contract, and then place the bids (and thus buy the NFT) from that contract. They can instead choose which wallet they want to use with the bot. This makes sense because the user doesn't have to transfer NFTs and funds back and forth between their own wallet and the smart contract, and if they are farming an airdrop, it is their own wallet that becomes eligible instead of the smart contract. To incorporate a smart contract to the project, I have deployed an NFT-contract to the Sepolia network and have token gated the bot instead.

Target audience

I believe there are three target audiences for this product; NFT collectors, NFT traders and airdrop farmers. The collector, simply because it will give them savings, traders because it can help them increase their profit margins, and airdrop farmers because they can be active users of Blur and OpenSea with the chance of making a profit, even without the airdrop.

Goal of the project

My goal for the six week project was to have a working product for OpenSea, which can later be expanded to include Blur as well. The most important feature is automatic bids based on user parameters, if there was time I wanted to add automatic cancellation of bids as well. I did not find this to be an important feature after all, since the bids are only valid for 15 minutes.

I want the user to be able to select which collection they want to bid on, how much they are willing to pay for an NFT, and how many they are willing to buy.

This will benefit the user because they can save time while buying NFTs at a discount, and if someone bids over their bid but not higher than the user is willing to go, the bot will overbid for them as well.

Tech stack

The frontend is made with React as that is the frontend library I am most familiar with. The functions that communicate with the smart contract are written in JavaScript and the smart contract for the NFT collection is written in Solidity. Since it's a simple product I believe I will be able to make the frontend easy enough to understand so that a user manual won't be necessary.

GitHub link

https://github.com/KristofferGW/nft-autobidder

Technical flow

When the user has bought WETH and approved OpenSea's smart contract to spend it, they can get started with the bot.

Once they connect to the bot's frontend, a function call will be made to the smart contract I deployed with an NFT collection to check if they hold the NFT. If they do, they will be able to proceed.

The first step is to find the collection they want to bid on and copy the collection slug from the URL, and paste that into the form in the bot frontend. They then enter the maximum they are prepared to bid for the NFT, and the minimum difference they want between their bid and the floor price of the collection, and finally they click "Start bot".

When "start bot" is clicked, the bot will fetch the current floor price and the highest bid for the collection at the moment, through OpenSea's API. If the user's max bid is lower than the current floor price but higher than the current max bid, it will place a bid that is 0.0001 WETH higher than the current max bid, also through OpenSea's API. The bid will be valid for 15 minutes, and after those 15 minutes the bot will repeat the process of checking if it should place a bid or not again.

This will go on until they click "Stop bot", which will trigger another API call to the node.js backend to stop the bot from placing bids.

Legal considerations

As the token is not intended to be traded on the secondary market, but simply used to control who uses the bot. And there is no promise of financial gains, I don’t see how this would fall under MiCA or any other regulation.

Reflection

I have reflected on a few things to enhance the user experience, which I will most likely build out after this project is complete, as I intend to use the bot. The first thing is that I would like more meaningful messages in the frontend. Like when a bid is placed, how much it was placed for and what the current top bid, and floor price are.

I would also like the user to be able to connect with their preferred wallet, instead of loading the .env with their private key. Both for ease of use and security reasons.

The last thing is that I would like to add a function call in the frontend, so that the user doesn’t have to visit OpenSea in order to approve WETH for trading.