Brian Prisbe

Game Theory in Computer Science

Dr. Vora

5/4/2022

Final Project Report

Link to GitHub: https://github.com/Prisbe/VickreyAuctionOnTheBlockChain

Introduction:

My project was the implementation of a Vickrey auction on the Ethereum Blockchain to sell an NFT. The auction is accomplished via a smart contract which is a set of instructions that is deployed on a decentralized blockchain.

Tools, Languages, and Frameworks Used:

- **Solidity**

  Solidity is an object-oriented programming language for implementing smart contracts on the Ethereum Blockchain.

- **Python**
  Python was used for compiling, deploying, and interacting with smart contracts on the Ethereum blockchain.
- **Brownie**

  Brownie is a python-based development and testing framework for smart contracts on the Ethereum Virtual Machine. It is built off the web3.py library. Brownie makes interacting with smart contracts and the blockchain simple.

- **Ganache**
  Ganache is a tool that allowed me to create a local Ethereum Blockchain to use for development and testing of my smart contracts. Ganache would automatically create fake Ethereum users with addresses and Ethereum. Ganache also showed all the blocks of the blockchain and all the transactions that were made upon the blockchain.
- **Rinkeby Test Network**

  Rinkeby is also a test network used for development and testing, but unlike ganache, Rinkeby is a fork of the main Ethereum network, so interaction on Rinkeby would be the exact same as on the mainnet.

- **Etherscan**
  Etherscan is a block explorer and analytics platform for the Ethereum mainnet. This website allows for easy access to blockchain data. Also, etherscan allows you to interact with smart contracts from within the website. If you want to keep up with my Ethereum transactions or send me ETH, just look me up on etherscan using my ENS name of *Prisbe.eth*.

GitHub Files Explained:

- Contracts/Sealed_Bid_Auction.sol
    - This is the smart contract code written in the solidity language. This code defines all the variables and functions that define the Vickrey auction.
    - This is the code that was be deployed to the Ethereum blockchain and would be given its own Ethereum address.
    - Transactions are made upon this smart contract by using the contract's address.
- Scripts/sealed_bid_sim.py
    - This is the python script that was shown during the presentation.
    - This script used the Ganache local blockchain.
    - The code would deploy a contract to the blockchain, have the owner sell the NFT via Vickrey auction, have bidders bid on the NFT, calculate results, and then winner of the auction paid for the NFT causing for ownership to be transferred to the winner.
- Scripts/helpful_scripts.py
    - This file defines many functions that can be used to call the different functions of the smart contract.
    - These functions were made because it made creating other scripts easier since I could just simply call these functions.
- Scripts/deploy.py
    - This is a simple function that was used to deploy a contract to the local Ganache blockchain.
- Scripts/deployToRinkeby.py
    - This is a function that was used to deploy a contract to the Rinkeby Test network.
    - This function is different than deploy.py since it needs to use our real Ethereum address and also allows Rinkeby to validate and show our raw solidity code on the blockchain.
- Scripts/no_bidder_auction.py
    - This is a script showing what would happen when no one bidded on the NFT.
    - The code starts with deploying a contract, the owner starting the auction, results calculated, and then the owner would have the ownership of the NFT reverted back to them since no one bidded.
- Scripts/testing_functions.py
    - This was a simple set of functions that I was using during development just to make sure that smart contract interaction was performing as expected.
- Brownie-config.yaml
    - This was just a configuration file that brownie used to pull my default network, private keys, and environment variables.

What I learned:

- **Blockchains**
    - I learned a lot about more about how blockchains actually works. I understood how transactions are secure and validated by using hash algorithms and digital signatures.
    - The website that visually showed me how the blockchain worked via hashing and digital signatures will be linked below:
        - Blockchain Demo: https://andersbrownworth.com/blockchain/
    - I learned more about how blocks are mined via Proof of Work and more about how Ethereum plans to change to Proof of Stake in the future.

- **Similarities and Differences between Bitcoin and Ethereum**
  - I learned the differences between two of the most popular cryptocurrencies out during time of writing.
  - The biggest difference between the two is that Ethereum is more than just a cryptocurrency and we see that with the implementation of smart contracts.
  - Ethereum allows us to create decentralized and tamper-proof applications and financial contracts.
- **Ethereum Development**
  - Mostly what I learned about is how to develop on the Ethereum blockchain.
  - I learned how to create, deploy, and interact with Solidity Smart Contracts.
  - I learned how to use Python and Brownie to automate my interaction with smart contracts.
- **Conclusion**
  - Overall, this project has opened my eyes to the endless possibilities that blockchain technologies allow for. I am very excited that the future of the internet will be Web3, and that infrastructure will become decentralized. As a cybersecurity professional, I know that this will also come with a lot of security risk, so I hope to continue being involved with blockchain technologies to protect and secure the network.

References

Collins, Patrick. [FreeCodeCamp.org]. (2021, September 9). Solidity, Blockchain, and Smart Contract Course – Beginner to Expert Python Tutorial [Video]. YouTube. https://www.youtube.com/watch?v=M576WGiDBdQ&t=19198s