Isha Patel

Running an Auction Using Blockchain

Introduction:

Blockchain is a fairly new technology that emerged with the introduction of the famous CryptoCurrency BitCoin. But there are many uses of blockchain besides just CryptoCurrency, for example, Walmart has a blockchain that allows them to track their produce and the United States Postal Service has placed a patent to use blockchain as an identity verification measure. Blockchain is used for tasks as simple as securing databases to as complicated as monitoring carbon offsets. It can also be used to manage an auction and would be able to safely transfer money, keep identities of bidders secure, and keep an accurate ledger of the bids.

A blockchain is basically a permanent digital ledger of transactions. Blockchains are decentralized meaning that the network has no central authority. That way there is not just one authority deciding what is true and altering the ledger for their own benefit. These ledgers are immutable; it is essentially impossible to change the ledger.[1] This is because of the chained blocks in the blockchain. Each block's hash, "the output of a mathematical function"[2], requires its previous block's hash.

Blockchains are also very secure while being transparent at the same time. They are transparent in the sense that the ledger is visible to everyone in the network, but it is secure and

TxHash	Block	Age	From		To	Value	[TxFee]
0x2d055e4585ae2a...	5629306	16 secs ago	0x003e3655090890...		0x2bdc9191de5c1b...	0,004741591554641 Ether	0.000294
0xb4d37c791ff4cde...	5629306	16 secs ago	0x6c3b4faf413e0e4...		0xf14cb3acac7b230...	0,744767225 Ether	0.000294
0x9979410dcb5f4c...	5629306	16 secs ago	0x99bcd75abbac05...		0x2d42ee86390c59...	0,016294 Ether	0.000294
0x189c4d4aae09be...	5629306	16 secs ago	0x175cd602b2a1e7...		0xd39681bb0586fb...	0,01 Ether	0.000294
0xda0e9bbb11fb77...	5629306	16 secs ago	0x73a065367d111c...		0x01995786f14357...	0 Ether	0.00150007
0x6be498fafad9acb...	5629306	16 secs ago	0xa3eb206871124a...		0x8a91cac422e55e...	0,029594 Ether	0.000294

Figure 1: This is a sample of an Ethereum (IDE for blockchain smart contracts) ledger.

private in the sense that a user's information is not shown. Instead, their public addresses are shown. For example, in figure 1 all of the transactions can be seen but the identities of the users are represented as their public addresses. Blockchains are also very secure in storing information because, as mentioned above, they are impossible to alter. It would take 51% of the servers to alter a block.[3] Creating a fake block can be easily spotted by the network unless 51% majority is in on the fraudulent activities and verifies the block. But the 51% attack is very very rare in large networks.[3]

As mentioned above, blockchains can be used for a lot of functions including auctions. Auctions deal with sensitive information, such as large amounts of money and the identities of the people with large amounts of money. There are many types of auctions; the four basic types of auctions are English auctions, Dutch auctions, first-price/sealed-bid auctions, and second-price/sealed-bid auctions, also known as Vickrey auctions.[4] The first two auction types are open, meaning that all the people know what each other's bids are, however, the latter two are sealed-bid, where each person places their bid in an envelope. Also, only the Vickrey auction is a second-price auction, meaning the highest bidder gets to pay the second-highest bid.[4] In the first-price auctions, the highest bidder pays the price they bid, but in the last auction type, the second-bid auction, the highest bidder gets to pay the second-highest bid. [4]

The goal is to build a smart contract for a blockchain-based English auction with one aspect of the Vickrey, which allows the highest bidder to pay the second-highest bid. The goal is also to learn more about blockchain and how it works.

More research was conducted to understand blockchains, the IDE called Ethereum that is well known for running smart contracts, and the programming language Solidity. After a base auction code was required, changes were made to add the extra features to the auction. Then the code was tested to find any glitches and to make sure that the number of ethers was correctly being subtracted and returned.

Procedures:

The project was built using Solidity which is an object-oriented programming language used to write smart contracts.[5] The project was built in Remix which is the IDE of a blockchain-based platform called Ethereum.[6][7]

Open-sourced resources were imported into the IDE to create the base code of blockchain-based auction that was used to set a foundation in order to add or remove features of the auction.[8] The open-sourced code was run to see the basic functions of the auction and it was noticed that an address or account was not being given to the seller. However, the reason for this was just human error. The rest of the code ran without any difficulty, however, in order to implement the Vickrey auction component, some changes were made.

A new variable named “previousBid” was created and it was set to another variable, “latestBid”, which was equal to the most frequent bid. The creation of this variable allowed for a

place to store the previous bid which would be necessary when the last bidder has to pay the auction of the previous bid. This variable was placed right before the value of the new bid was set equal to “latestBid”.

```
previousBid = latestBid;
latestBid = msg.value;
```

Figure 2: This is what the code to set up “previousBid” looked like.

Another variable named “count” was also added; its purpose was to keep track of the number of bids placed. Determining the count was important because the addition of the code from above allowed a sole bidder to not pay any ethers, which are the currency of Ethereum. When the auctioneer opened an auction, the count was set to zero and whenever a bid was placed, the count increased by one. So if the count was equal to one, then the sole bidder paid what they bid, but if there were more than one bidder, then the last bidder paid the second-highest bid.

```
constructor() public {
    auctioneer = msg.sender;
    count = 0;
}
.
.
.
    previousBid = latestBid;
    latestBid = msg.value;
    latestBidder = msg.sender;
    count = count + 1;
```

Figure 3: This is what the code to set up the count looked like. It was also used to

help determine the amount of ethers a bidder paid.

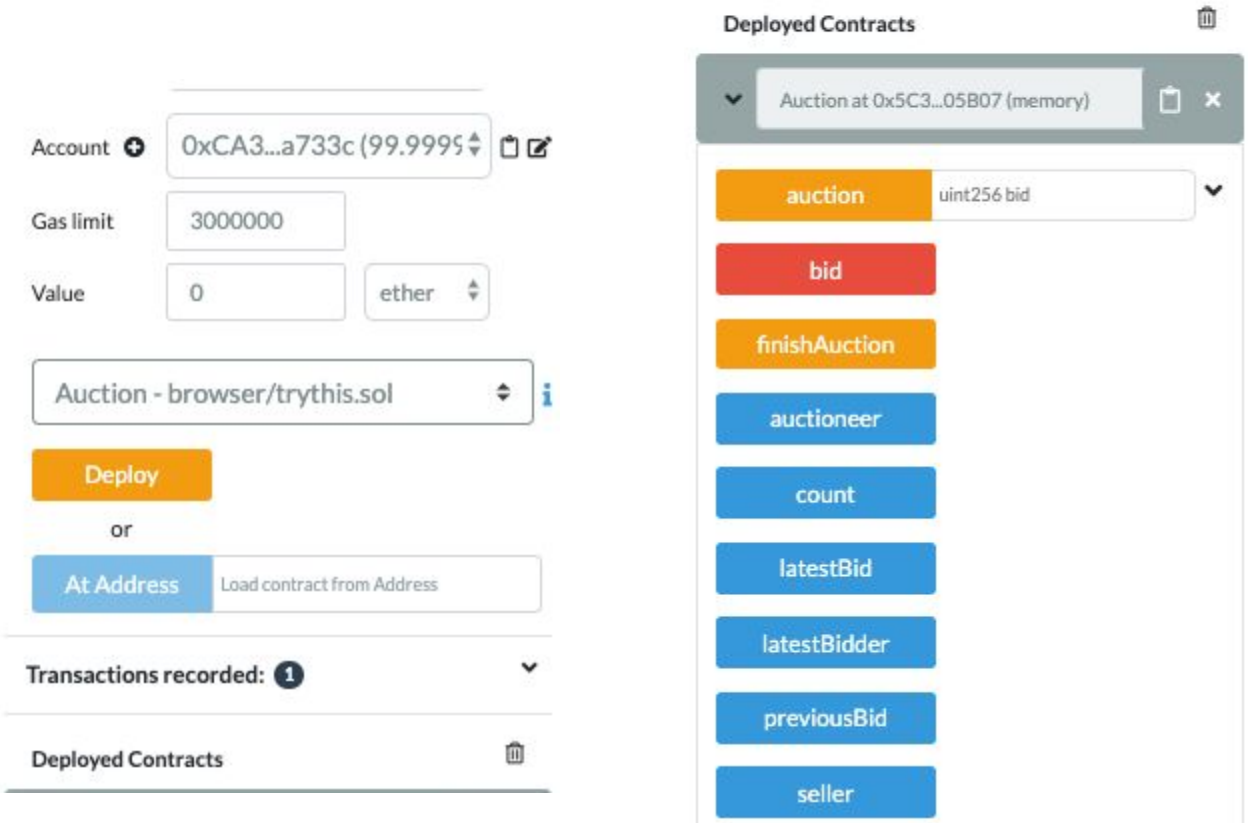

Figure 4: This is the console that helped run the auction. Each blue "button" in the console were the variables declared in the code. If those buttons were pressed, then the address or value currently assigned to that respective variable could be seen. The orange "auction" button assigned the seller and the minimum bid, and the orange "finishAuction" button ended the auction. The red "bid" button sets a bid.

There are already five addresses or accounts preset in the IDE, each with 100 ethers to spend. An address was selected from the account dropdown to open the auction, and then the "deploy" button is pressed. The selected address becomes the auctioneer and the auctioneer was

the only address that was also able to close the auction.

After the auction was opened, another address was selected to become the seller, a minimum bid value was entered into the space beside the "auction" button, and the button was clicked to set the seller and the minimum bid. The rest of the addresses were the bidders. To place a bid, an address had to be selected and a value, which was greater than the value of the previous bid, was entered into the value box and the red "bid" button had to be clicked; this cycle of bidding continued until the bidders would not go further. Then as mentioned above, the auctioneer was selected to close the auction by pressing the orange "finishAuction" button.

Results:

The auction was run multiple times to test if the right amount of ethers were being transferred to the right accounts. If there were no bids cast, then after the auction closed, the number of ethers each address stayed the same. If there was only one bid, then that bidder would have paid the amount they bid because there was no previous bid. However, if there were two or more bidders, then the highest bidder would have paid the second-highest bid like it is done at a Vickrey auction.

```
function finishAuction() restricted public {
    if(count == 1){
        seller.transfer(address(this).balance);
    }
    else{
        seller.transfer(previousBid);
        latestBidder.transfer(address(this).balance);
    }
}
```

Figure 5: This is the if/else loop that helped the bidders determine what they had

to pay.

Something else that was discovered while testing the auction was that the change in the number of ethers in each address's accounts changed after each bid instead of only at the end of the auction like most live auctions. Also, after a higher bid was cast, the ethers of the lower bid would return to the respective address and the smart contract itself stored the ethers that were bid. For example, if address two bid five ethers, those five ethers were stored into the smart contract's balance; the "(address(this).balance)" referred to the ether balance of the smart contract. Continuing with the example: then if address four bid ten ethers, then address two's five ethers were returned from the smart contract balance and address four's ten ethers were transferred to the smart contract.

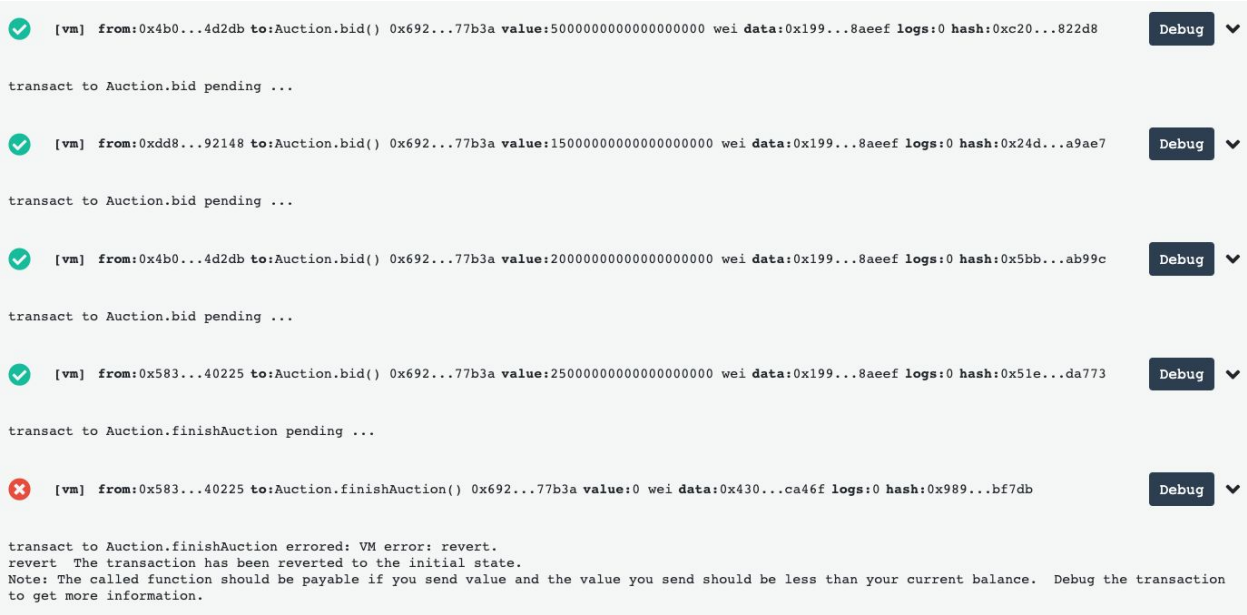

Figure 6: This is an example of a blockchain that was formed as the auction proceeded. Each green check represented an approved block and the red "x" represented a block that was not verified.

Figure 7: If a block is opened, then the contents or transaction history were seen.

A blockchain-like ledger was also being formed while an auction was running. The deployment of the auction, each of the bids placed, and the end of the auction transaction were all represented by individual blocks. As seen in Figure 6, each green check represented a block that was approved and the red "x" represented a block that was not approved. If a block was opened, then the information regarding that particular would be seen. For example, if a block representing a bid was opened, then the sending address, receiving address, amount of ethers being transferred, and other information were seen.

Conclusion:

Research was conducted to understand Ethereum and Solidity. After a base auction code was required, changes were made to add the extra features to the auction. The code was then run multiple times to make sure that the ethers were correctly being distributed.

The overall result of this project was that a smart contract was built for a blockchain-based auction that uses components of a minimum-bid auction merged with a component of a Vickrey auction. The auction lets the seller set the minimum bid and the highest bidder gets to pay the second-highest bid. Although most of the functions have to be manually selected, such as changing the bidder, the auction runs smoothly. In the end, the project was a success because it met the criteria set for it, which was to set up a functioning auction and add the extra Vickrey auction component.

Blockchain has countless possibilities, but for this specific blockchain-based auction, future research would include being able to use third-party timers to restrict the amount of time the bidders have to bid to automate the closing of the auction if said time runs out. This would give a more auction-like feel to the project. Another thing to possibly work on in the future is to format a front-end website for this auction where different people are the bidders and they decide the bids, instead of the bidders having to be manually selected.

Bibliography

[1] Rosic, A., & Blockgeeks. (2019, August 15). What is Blockchain Technology? A Step-by-Step Guide For Beginners. Retrieved from https://blockgeeks.com/guides/what-is-blockchain-technology/

[2] Coinmama.com. (n.d.). What is the Blockchain? Retrieved from https://www.coinmama.com/guide/what-is-the-blockchain

[3] Frankenfield, J. (2020, February 13). What Is a 51% Attack? Retrieved from https://www.investopedia.com/terms/1/51-attack.asp

[4] Fine, L. R., Small, K. A., Haring, J., Tollison, R., & Scully, G. W. (n.d.). Auctions. Retrieved from https://www.econlib.org/library/Enc/Auctions.html

[5] Solidity. (n.d.). Retrieved from https://solidity.readthedocs.io/en/v0.6.1/index.html

[6] Ethereum IDE. (n.d.). Retrieved from http://remix.ethereum.org/

[7] Ethereum. (n.d.). Retrieved from https://ethereum.org/

[8] Kok, A. S. (2019, April 3). Write A Simple Contract On Top Of Ethereum. Retrieved from https://medium.com/coinmonks/write-a-simple-contract-on-top-of-ethereum-92b543594e84