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.