

1. 목적

solidity를 이용하여 자동 입찰 시스템과 시간 연장 기능을 가진 경매 시스템을 만든다.

2. 주요 코드 설명

(1) attribute

```
//auto increasing bid value
uint autoIncreasingBidValue;

//auto increasing end time value
uint autoIncreasingTimeValue;

//available max bid value by address
mapping(address => uint) maxAvailableBid;
```

그림 1 attribute 추가

그림1과 같이 입찰 가격이 자동으로 증가되는 단위를 저장하는 autoIncreasingBidValue, 시간 연장 단위인 autoIncreasingTimeValue, 사용자별 최대 입찰 가격을 저장하는 maxAvailableBid attribute를 만든다.

(2) bid()메소드 – 자동 입찰 시스템

```
function bid(uint initial) public payable {
    //slide #7
    //write your code for bid function
    //max bid of user is msg.value
    require(msg.value>=initial,"max available bid should be greater than or equal to initial bid");
    require(now<=auctionEndTime,"Auction already ended.");
    require(initial>highestBid,"There is a higher bid than your initial bid.");
```

그림 2 bid 메소드1 – require()

먼저 그림2와 같이, 최대 입찰가가 초기 입찰가보다 같거나 크지, 경매가 마감되지 않았는지, 현재 경매에 등록된 최대입찰가가 사용자의 초기 입찰가보다 크지를 검사한다.

```
if(highestBid!=0){
    //if highestBidder can bid more
    if(maxAvailableBid[highestBidder]>msg.value + autoIncreasingBidValue){
        highestBid = msg.value + autoIncreasingBidValue; //auto bid more
        pendingReturns[msg.sender] = msg.value; //return all ether received from send
        pendingReturns[highestBidder] = maxAvailableBid[highestBidder] - highestBid;
    }

    //if highestBidder can't bid more
    else{
        maxAvailableBid[msg.sender]=msg.value; // set max available bid
        highestBid = maxAvailableBid[highestBidder] + autoIncreasingBidValue; // set
        pendingReturns[highestBidder]=maxAvailableBid[highestBidder]; //return all ether
        highestBidder=msg.sender; //change highestbidder to sender(msg.sender win)
    }
}
else{
    highestBidder = msg.sender;
    highestBid=initial;
    maxAvailableBid[msg.sender]=msg.value;
}
```

그림 3 bid() 메소드 – 자동 경매 시스템

그림3과 같이, highestBid가 이미 설정된 경우, 현재 highestBidder가 방금 입찰한 사람의 최대 가능한 입찰가보다 (0.1이나 1 ether 정도) 더 많이 낼 수 있는지 검사한다. 만약 그렇다면, 현재 highestbidder가 신청한 입찰가를 변경(방금 입찰한 사람의 최대 가능한 입찰가 + 0.1 or 1 ether)하고, 돌려줄 ether를 설정한다. highestBidder는 highestBidder가 설정한 최대 가능한 입찰가에서 입찰한 가격을 뺀 나머지를 돌려받는다. 방금 입찰한 사람은 전송한 ether를 모두 돌려받을 수 있다.

만약 더 많이 낼 수 없다면, 새로 입찰한 사람이 highestBidder가 된다. 입찰가는 이전 highestBidder의 최대로 가능한 입찰가에서 +(0.1 or 1 ether)이다. 이전 highestBidder는 전송한 ether를 모두 돌려받을 수 있다. 새로운 highestBidder는 설정한 최대 가능한 입찰가에서 입찰한 가격을 뺀 나머지를 돌려받는다.

(3) bid() 메소드 - 시간 연장

```
//Automatically extend the auction time if a bid is made
if(auctionEndTime - now <= 10){
    auctionEndTime += autoIncreasingTimeValue;
    emit AuctionExtended(autoIncreasingTimeValue);
}
```

그림 4 bid() 메소드 - 시간 연장

그림4와 같이 bid()함수가 호출되고 require()를 모두 만족시킨 후에, 마감 시간과 현재 시간의 차이가 10초 이하일 경우 마감 시간을 생성자에서 설정한 autoIncreasingTimeValue만큼 늘린다.

3. 동작 설명 및 로그

```
0x454...40000
{
  "uint256 initial": "1000000000000000000"
}
{}
[
  {
    "from": "0xD7ACd2a9FD159E69Bb102A1ca21C9a3e3A5F771B",
    "topic": "0xf4757a49b326036464bec61e419a4ae38c8a02ce3e68b10809674f6aab8ad300",
    "event": "HighestBidIncreased",
    "args": {
      "0": "0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2",
      "1": "1000000000000000000",
      "bidder": "0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2",
      "amount": "1000000000000000000"
    }
  }
]
300000000000000000 wei
```

한 사용자(0xAb...)가 초기 입찰가 1 ether, 최대 입찰가 3 ether로 신청한다.

```
{
  "uint256 initial": "2000000000000000000"
}
{}
[
  {
    "from": "0xD7ACd2a9FD159E69Bb102A1ca21C9a3e3A5F771B",
    "topic": "0xf4757a49b326036464bec61e419a4ae38c8a02ce3e68b10809674f6aab8ad300",
    "event": "HighestBidIncreased",
    "args": {
      "0": "0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2",
      "1": "2100000000000000000",
      "bidder": "0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2",
      "amount": "2100000000000000000"
    }
  }
]
200000000000000000 wei
```

다른 사용자(0x4B2...)가 최대 입찰가 2 ether로 신청하면, 0xAb가 신청한 경매의 입찰가가 2.1 ether로 증가된다.