**DCC Auction**

스마트 컨트랙을 이용한 가상경매 dApp

신권철

skc7401@gmail.com

Std ID: 9379920181

1. 개요

이 분산 어플리케이션(dApp)은 Metamask의 Ropsten Test Net에 배포된 스마트 컨트랙과 통신한다. 따라서 크롬이나 파이어폭스 브라우저를 이용해야 하며 각 브라우저에서 크롬 확장프로그램 혹은 파이어폭스 에드온을 통해 [metamask를 설치](http://metamask.io/)한 후 이용 가능하다.

2. 프로그램 구조

App은 세 가지 파일로 이루어져있다.

1. DCC\_Auction.html

애플리케이션의 UI를 구현하며 사용자의 입력을 전달하는 데 사용된다.

1. DCC\_Auction.js

Web3.js 라이브러리를 이용해 스마트 컨트랙과 통신을 하며 브라우저에 결과를 전달한다.

1. DCC\_Auction.sol

스마트 컨트랙을 구현한 파일

3. 실행

아래 웹페이지로 접속하면 바로 이용이 가능하다. 다만 앞서 언급했듯이 metamask가 설치돼있어야 하며 로그인 후 Ropsten Test Net으로 접속해야 한다. 먼저 화면 우측 하단에 Purchase Token을 통해 이더를 토큰으로 교환한 후 원하는 제품에 Bidding을 하면 된다.

<https://kwoncharles.github.io/Assignment_2/DCC_Auction.html>

4. 구현 코드 설명

**DCC\_Auction.sol**

컨트랙트를 구현한 solidity 파일

Variables

**struct bidder** {

address bidderAddress; // bidder의 주소

uint tokenBought; // bidder의 토큰 잔액  
}

* 경매 참여자의 정보가 저장될 구조체

**mapping (address => bidder) public bidders**

* bidder 구조체들이 저장될 KeyValue Store

**mapping (address => mapping (bytes32 => uint)) public usersBids**

* 각 bidder의 최근 bidding 값이 저장될 KeyValue Store

**bytes32[] public itemLis**t

* 경매에 나가게 될 아이템 목록

**uint public totalToken**

* 컨트랙의 토큰 발행 총량

**uint public balanceTokens**

* 현재 발행 가능한 토큰량

**uint public tokenPrice**

* 토큰 가격 – Wei로 환산

Contract functions

**function buy() public payable**

* 현재 컨트랙을 생성한 bidder의 토큰을 충전하는 함수(payable로 선언돼 msg.value 값을 불러올 수 있다.)

**function getHighestBids() public view returns (uint[])**

* 각 경매품의 가장 높은 입찰가를 반환하는 함수

**function getUserBids() public view returns (uint[])**

* 각 경매품에 대해 현재 컨트랙을 생성한 bidder의 최근 입찰가를 반환

**function bid(bytes32 itemName, uint tokenCountForBid) public**

* 현재 컨트랙을 생성한 bidder가 입찰을 할 때 호출하는 함수. 아이템 이름과 입찰가격(토큰)을 인풋으로 받는다. 입찰가격이 기존의 최고입찰가보다 크거나 bidder의 토큰잔액보다 적어야 입찰이 진행된다.

**function getItemIndex(bytes32 item) public view returns (uint)**

* 경매품의 이름을 입력받아 리스트 인덱스를 반환하는 함수

**function getItemsInfo() public view returns (bytes32[])**

* 현재 컨트랙에서 경매가 진행중인 상품의 리스트를 반환하는 함수

**function getTotalToken() public view returns (uint)**

* 현재 컨트랙의 토큰 총량을 반환하는 함수

**function getBalanceTokens() public view returns (uint)**

* 현재 컨트랙에서 발행 가능한 토큰량을 반환하는 함수

**function getTokenPrice() public view returns (uint)**

* 토큰의 단가(Wei)를 반환하는 함수

**function getTokenBought() public view returns (uint)**

* 현재 컨트랙을 생성한 bidder의 토큰 잔액을 반환하는 함수

**DCC\_Auction.js**

Web3.js 라이브러리를 이용해 스마트 컨트랙과 통신하기 위한 JS 파일

Variables

**const contractAddress**

* 현재 스크립트 코드와 연결할 스마트 컨트랙 주소

**const abi**

* 현재 스크립트 코드의 abi

**let auctionContract**

* abi가 삽입된 스마트 컨트랙 객체. Initialize 역할

**let auction**

* 스마트 컨트랙 주소와 연결된 스마트 컨트랙 객체

**let accountAddress**

* 현재 스크립트 코드를 실행하고 있는 노드의 주소

**let currentEtherBalance**

* 현재 스크립트 코드를 실행하고 있는 노드의 이더리움 잔액

**let currentTokenBalance**

* 현재 스크립트 코드를 실행하고 있는 노드의 컨트랙 토큰 잔액

**let tokenPrice**

* 연결하고자 하는 스마트 컨트랙의 토큰 단가(Wei)

**const itemNames**

* 연결된 스마트 컨트랙에서 진행하는 경매 아이템 목록(utf-8)

**const itemNamesBytes32**

* 연결된 스마트 컨트랙에서 진행하는 경매 아이템 목록(bytes32)

Functions

**function startApp()**

* 현재 스크립트가 실행될 페이지가 로드되면 실행되는 함수로 스마트컨트랙 객체 생성, 노드 주소 로드, 아이템 목록 로드 등 기본적인 세팅을 진행하는 함수

**function getLink()**

* 현재 스크립트를 실행하는 유저의 노드 주소를 불러오는 함수

**function getEther()**

* 현재 스크립트를 실행하는 유저의 이더리움 잔액을 반환하는 함수

**function getToken()**

* **스마트컨트랙 함수 호출**
* 현재 스크립트를 실행하는 유저가 스마트 컨트랙에서 보유하고 있는 토큰량을 반환하는 함수

**function getTokenInfo()**

* **스마트컨트랙 함수 호출**
* 스마트 컨트랙의 토큰 총량, 현재 발행량, 토큰 가격, 토큰 잔액을 반환

**function getHighestBids()**

* **스마트컨트랙 함수 호출**
* 각 경매품들의 최대 입찰가를 반환

**function getMyBids()**

* **스마트컨트랙 함수 호출**
* 각 경매품들에 대한 현재 노드의 최신 입찰가를 반환

**function bidForProduct(i)**

* **스마트컨트랙 함수 호출**
* 입력받은 아이템에 대한 입찰을 진행하는 함수. 입찰가는 사용자가 기입한 값을 element ID를 통해 불러온다.

**function buyTokens()**

* **스마트컨트랙 함수 호출(payable)**
* X현재 유저가 이더리움으로 토큰을 사게 해주는 함수.