110 學年 第 2 學期 金融科技安全 期末專題報告 日期:2022/6/23

(專題組員為 1 至 2 人為限)

班級: 資工三 學號: B0843020 姓名: 吳宥俞

班級: 資工三 學號: B0843001 姓名: 柯維哲

1. 區塊鍊智能合約應用程式開發 (共 1 題，100 分，滿分 100 分)

請以課程中所教授智能合約範例為基礎，構思一 Ethereum 智能合約應用，並透過 Remix IDE + Metamask 部署於 Test Network。

請於 2022/6/23 23:59 前透過數位學習園區繳交報告及程式碼

A. 描述本專題的題目、動機、目的與架構。

題目:具有加權投票機制的智能合約

動機:「投票」是公民展現自由、公平、公正的表現，在民主國家中，投票環節是不可或缺的公民活動。然而每次舉辦大型投票活動，例如總統大選時，政府總是需要**花費大量的人力、精力、經費**，來確保整個過程的成功。其實在近幾年，「**數位投票**」的想法就有出現過，但是**總是因為可能會有作弊、黑箱的疑慮，且很難查證，故數位投票的方法一直無法實現**。本專題利用區塊鏈的特性，撰寫一套智能合約，使整個投票的環節擁有其無法竄改、公開透明、不透過第三方經手的特性，使公民能夠完全信任開票結果。

圖 1、2014 年選舉經費預估，費用皆破億

若是使用本專題的投票方式的話，政府只需要對每個經實名制認證後的帳戶發送投票時所需要的手續費給公民，就可以**省下數十億的花費**。且實地投票時需要返回自己的戶籍地，會造成交通阻塞、民眾群聚，利用線上投票的方法即可不因疫情影響仍正常投票，也可以使人民用非常方便的方式參與投票。

目的:撰寫並部屬一份智能合約，使公民能利用合約進行公平、公正、公開的投票

情境描述:設計一簡易的投票系統智能合約，其合約需發起人(leader)的 address、候選人的資料，在合約發行後，發起人需要賦予經過

認證的人投票權，並讓有投票權的人進行投票，最後投票結束後，需要顯示當選人的姓名及最高票的票數。架構圖如下所示。

架構:需具有 modifier 功能:

1.判斷只有發起者(leader)才可以執行開票的功能

需具有 function 功能:

1.發起人能夠賦予人投票權
2.投票功能
3.開票功能
4.初始化功能(輸入候選人資料)

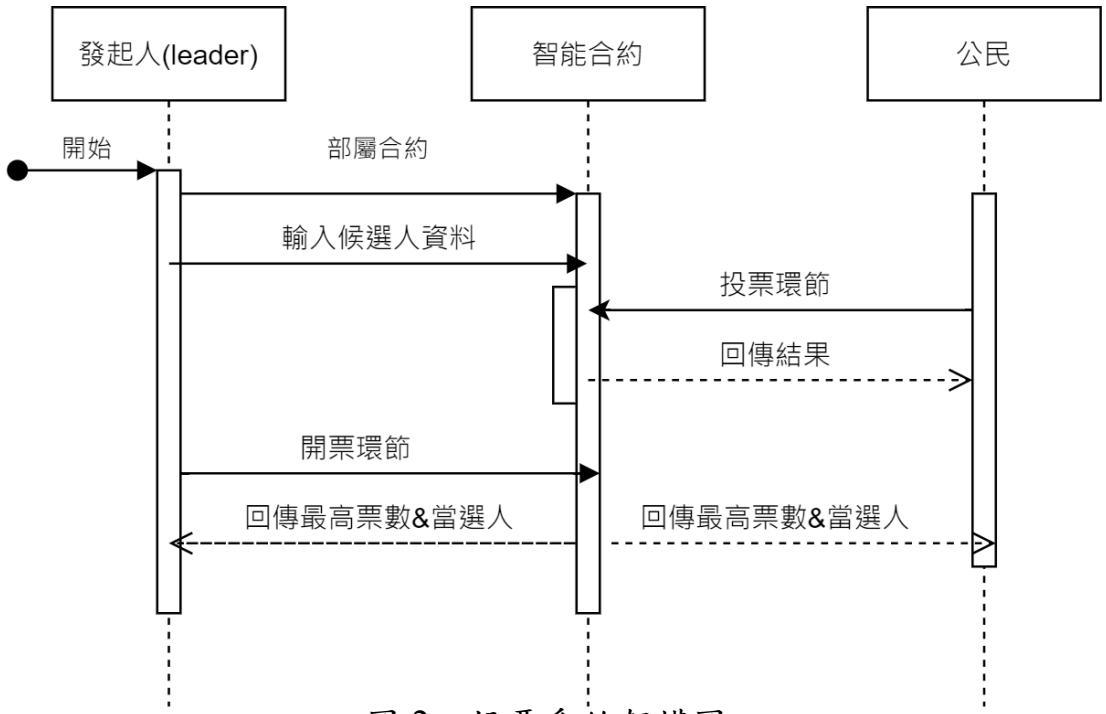


圖 2、投票系統架構圖

B. 解說本專題程式的開發環境、關鍵程式碼。

版本

使用的 solidity 版本為>=0.4.22 <0.6.0。

```
pragma solidity >=0.4.22 <0.6.0;
```

圖 3、solidity 版本

首先，將候選人的姓名、票數利用 struct 打包起來，還有投票者的投票權重、是否投過票、投票投給誰的資訊也利用 struct 打包起來。

```
struct Proposal {
    string name;   //候選人的姓名
    uint voteCount;//候選人的投票計數
}

struct Voter {
    uint weight;     //投票人的權重
    bool voted;      //投票人是否投過票了
    uint votedProposal; //投票人投給誰
}
```

圖 4、宣告 struct 的內容

接著利用 mapping，追蹤投票者的位址，方便利用其位址去找尋其他 struct 裡面的資料。

```
mapping(address => Voter) voters; //利用投票者的地址追蹤Voter裡的struct的資訊,以利追蹤
```

圖 5、mapping 追蹤投票者的資訊

接下來介紹各個 function

首先是建構子，先將發起這份合約的人視為發起人(leader)，並且將發起人投票權重定為 1，表示發起人也可以投票，並且在建構子的地方將所有候選的資訊，包括姓名、票數這些變數加進 list 內。

```
//建構子,在建構的時候順便紀錄發起人(leader)是誰
constructor() public {
    leader = msg.sender; //發起投票的人
    voters[leader].weight = 1; //發起投票的人的投票權重(發起人也可以投1票)

    //把所有候選人加進提案列表
    proposals.push(Proposal({
        name: "阿糖",
        voteCount: 0
    }));

    proposals.push(Proposal({
        name: "力量人",
        voteCount: 0
    }));
}
```

圖 6、constructor 建構子

接下來是 giveRightToVote()，該 function 能夠給定一個人投票權，在投票的時候要有投票權才能夠投票，以防止一人多投票等等的情況發生，首先先確認使用這個 function 的人是發起者(leader)，接著檢查輸入的位址的人，看看該人是還沒投票且還沒有權利投票的人，若都符合的話，就將該人的投票權重設為 1，使該人可以進行投票。

```
//由leader指定那些人可以投票
//確保每個人只能用一個帳號投票
function giveRightToVote(address voter) public {
    require(msg.sender == leader);          //確認使用這個function的人是leader
    require(voters[voter].voted == false);  //確認要被授權投票權利的人是還沒有投票的人
    require(voters[voter].weight == 0);     //確認要被授權投票權利的人是沒有投票權利的人

    voters[voter].weight = 1;               //以上都有符合的話,把這個人的投票權重設為1
}
```

圖 7、giveRightToVote()方法

接下來是投票的 function，若有公民想投票，就會使用到這個 function，其內容會先檢查該公民是否還沒投過票、且是有權利投票的人，如果是，就將它輸入的投票結果加在該候選人上，並將該位址設定為以投過票。

```
//投票環節
function vote(uint proposal) public {
    Voter storage sender = voters[msg.sender];  //用storage修飾這個變數讓這個投票者的資訊會被記錄在區塊鏈上
    require(sender.voted ==false);              //確認投票者是還沒有投過票的人
    require(sender.weight > 0);                 //確認投票者是有權利投票的人

    sender.votedProposal = proposal;            //將投票者投給誰的結果記錄在區塊鏈上
    sender.voted = true;                        //將投票者設為已投票

    proposals[proposal].voteCount += sender.weight;//把候選人的票數加上去
}
```

圖 8、vote()方法

最後是開票的 function，在任何期間，都可以去查詢當下的開票結果，內如為紀錄貴高票的票數，以及其候選人。

```
//開票環節
//要在資訊欄位看到目前的最高票數,還有候選人的姓名
function showWinningProposal() public view returns (uint winningProposal , string memory winningProposalName) {
    uint maxVote = 0;
    //利用for迴圈找出所有提案得票數最多的
    for(uint i = 0; i < proposals.length; i++) {
        if(proposals[i].voteCount > maxVote) {
            maxVote = proposals[i].voteCount;
            winningProposal = i;                    //紀錄最高票
            winningProposalName=proposals[i].name;  //紀錄姓名
        }
    }
}
```

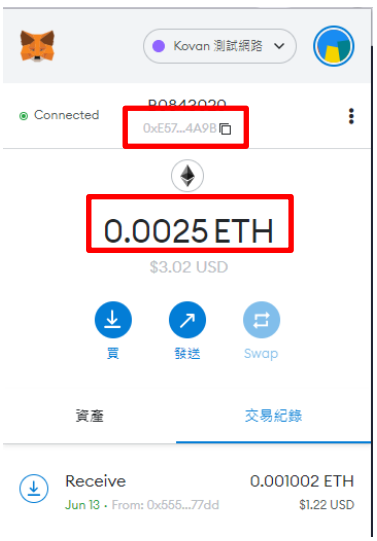
圖 9、showWinningProposal()方法

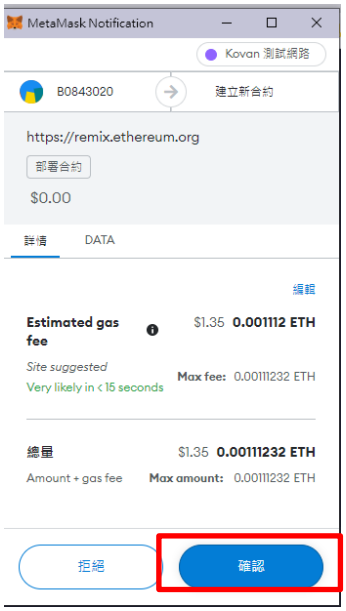
C. 呈現本專題程式的執行結果。

在本專題中，我們利用 3 個錢包地址，分別代表發起人、2 位公民:

發起人地址: 0xE5760b44D96C8ECc851d86Af8c0A714F54524A9B
公民 1 地址: 0xA478a6d801eFbD2cc50d08cA779CB5a5eB00708F
公民 2:地址: 0x7ce6827C99088367a7b8779367588bf267b2c04f

由發起人(4A9B)部屬合約:





部屬完後，付出一些 gas fee，成功部屬合約。

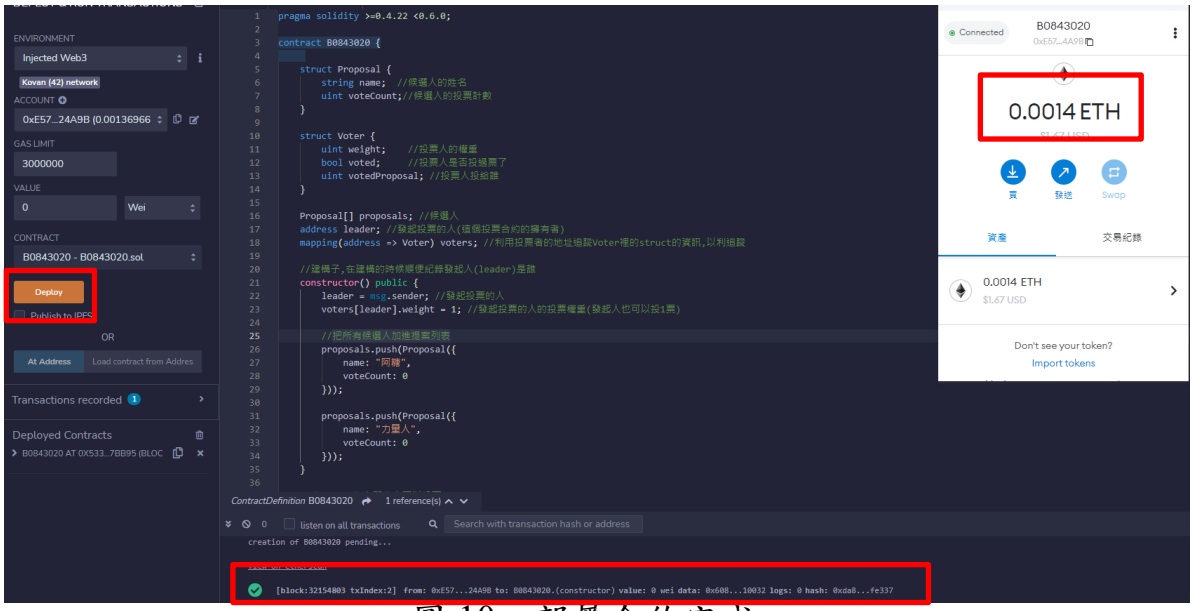


圖 10、部屬合約完成

接著可以看到下方有剛剛介紹的數種 function

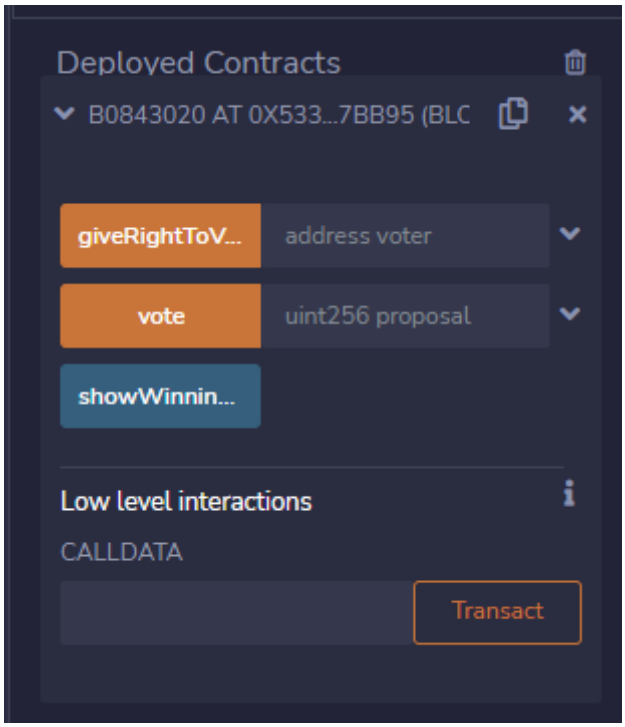


圖 11、數種 function 供操作

可以看到，若不是由發起人點選賦予投票權的 function，合約就會報錯。

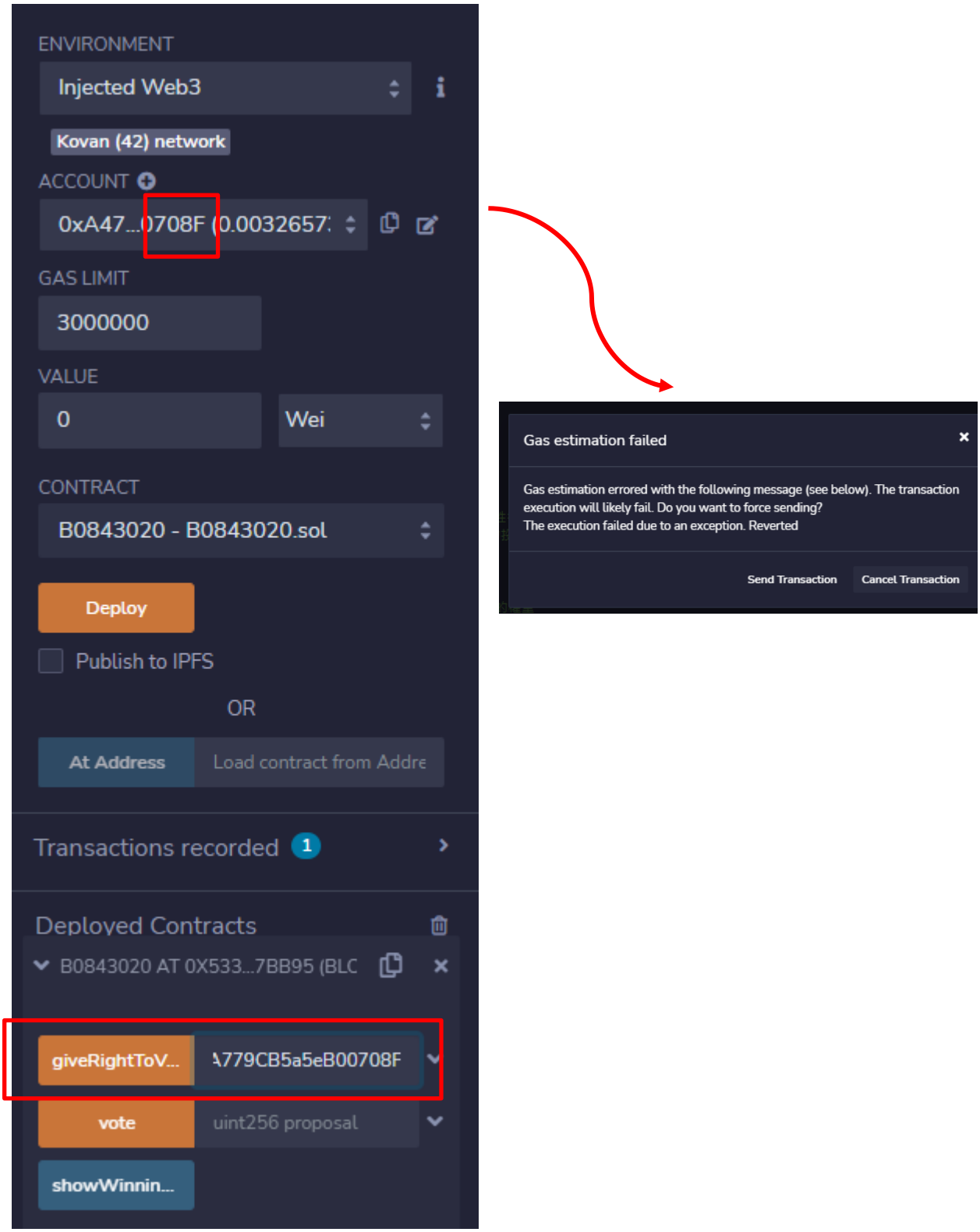


圖 12、確認 function 是否為發起人點選

切換到發起人(4A9B)後，就可以正常使用該 function

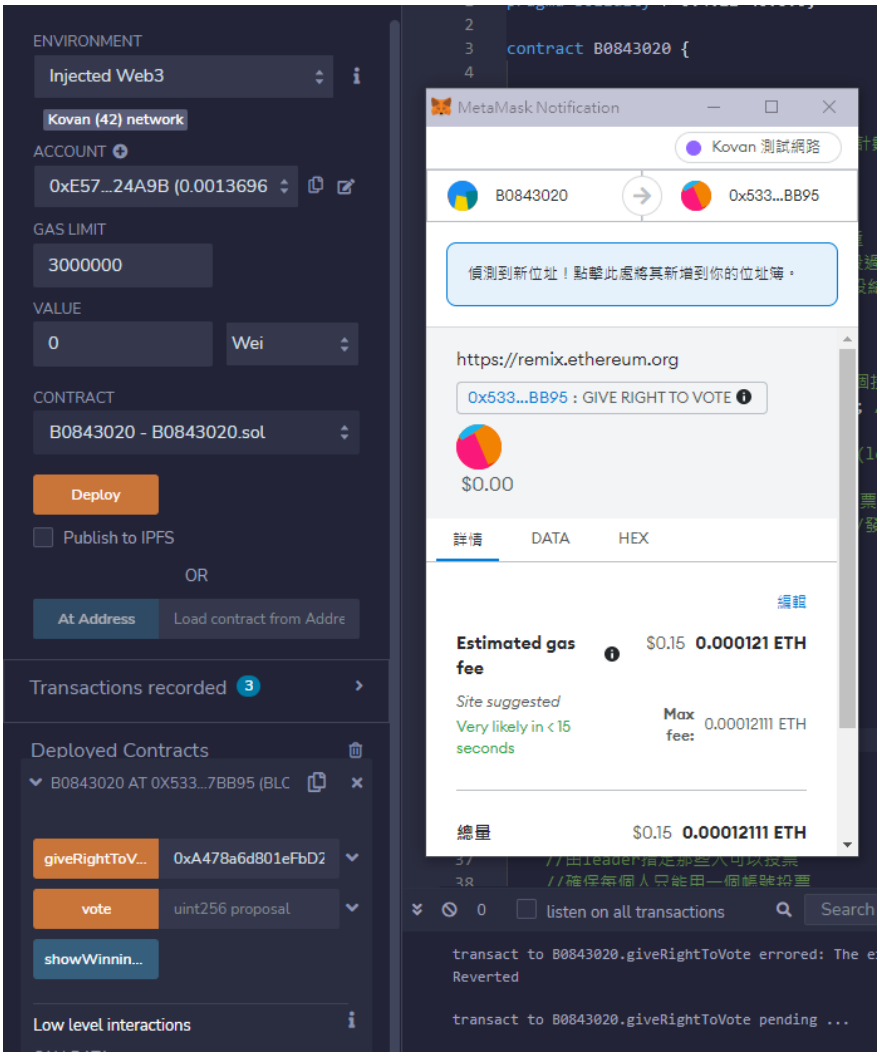
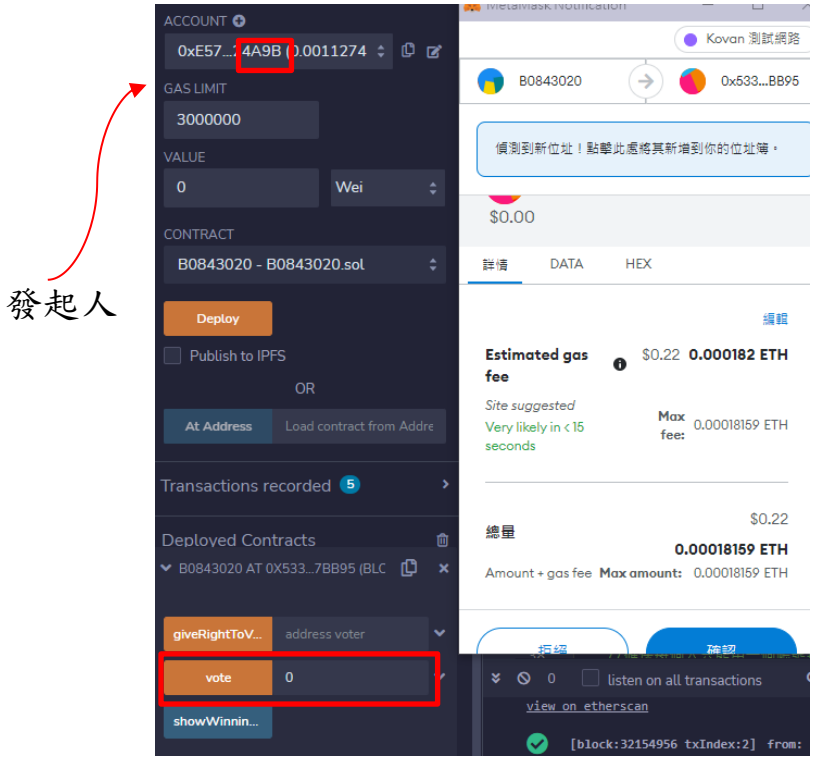


圖 13、function 正常運作

將公民 1(708F)及公民 2(c04f)都賦予其投票權後，即可開始投票，包含發起人，總共 3 票，以下範例為發起人投給”阿糖”候選人，公民 1 及公民 2 投給”力量人”候選人。



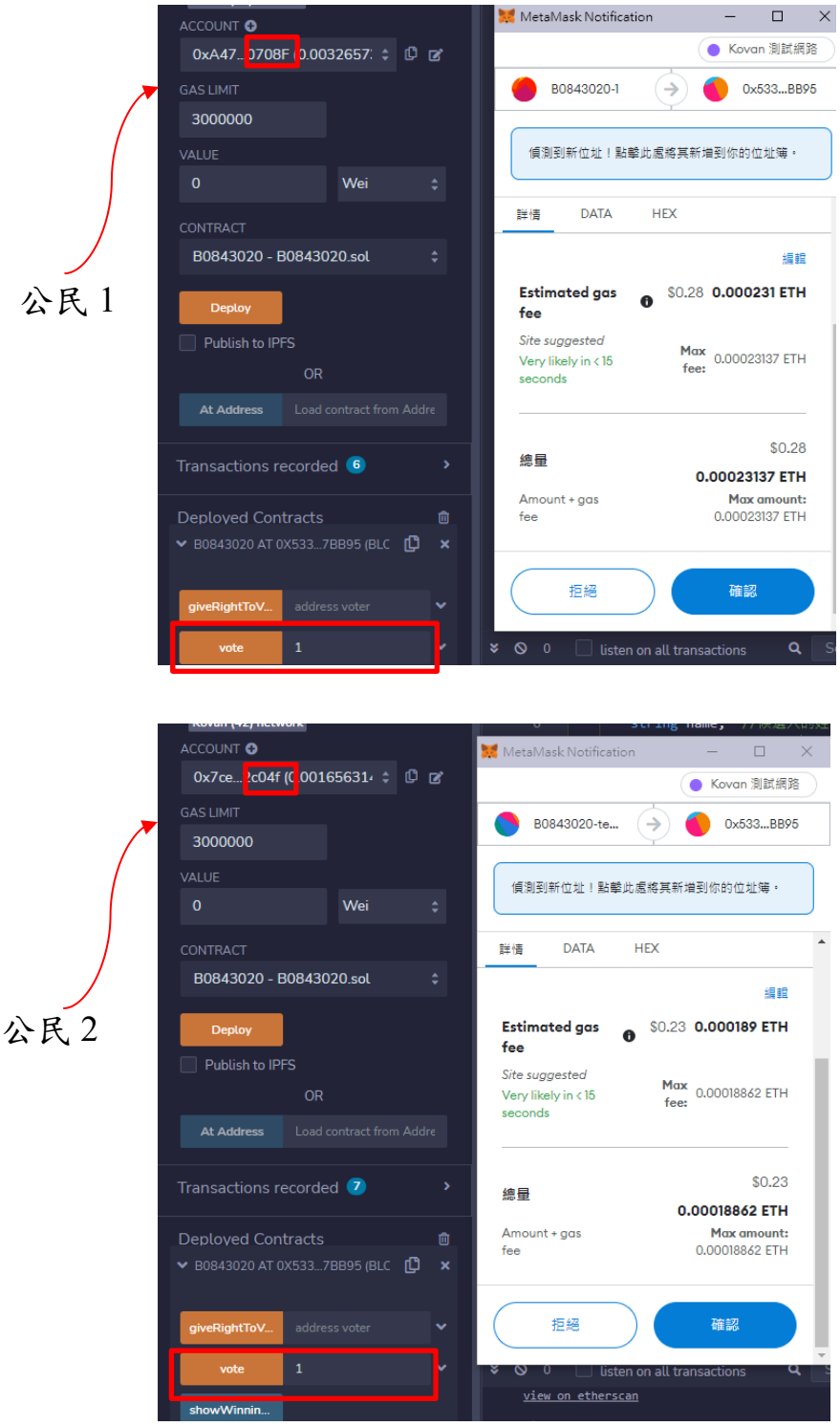

公民 1
ACCOUNT
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
B0843020 - B0843020.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Addre
Transactions recorded 6
Deployed Contracts
B0843020 AT 0X533...7BB95 (BLC
giveRightToV...
address voter
vote
1
MetaMask Notification
Kovan 測試網路
B0843020-1
0x533...BB95
偵測到新位址！點擊此處將其新增到你的位址簿。
詳情
DATA
HEX
編輯
Estimated gas fee
$0.28
0.000231 ETH
Site suggested
Very likely in < 15 seconds
Max fee:
0.00023137 ETH
總量
$0.28
0.00023137 ETH
Amount + gas fee
Max amount:
0.00023137 ETH
拒絕
確認
listen on all transactions
公民 2
ACCOUNT
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
B0843020 - B0843020.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Addre
Transactions recorded 7
Deployed Contracts
B0843020 AT 0X533...7BB95 (BLC
giveRightToV...
address voter
vote
1
showWinnin...
MetaMask Notification
Kovan 測試網路
B0843020-te...
0x533...BB95
偵測到新位址！點擊此處將其新增到你的位址簿。
詳情
DATA
HEX
編輯
Estimated gas fee
$0.23
0.000189 ETH
Site suggested
Very likely in < 15 seconds
Max fee:
0.00018862 ETH
總量
$0.23
0.00018862 ETH
Amount + gas fee
Max amount:
0.00018862 ETH
拒絕
確認
listen on all transactions
view on etherscan

圖 14、投票環節

結束後，點選 showWinningProposal()查看目前最高票的人是誰，可以看到是第 1 位(阿糖為第 0 位)候選人—“力量人”為最高票數。

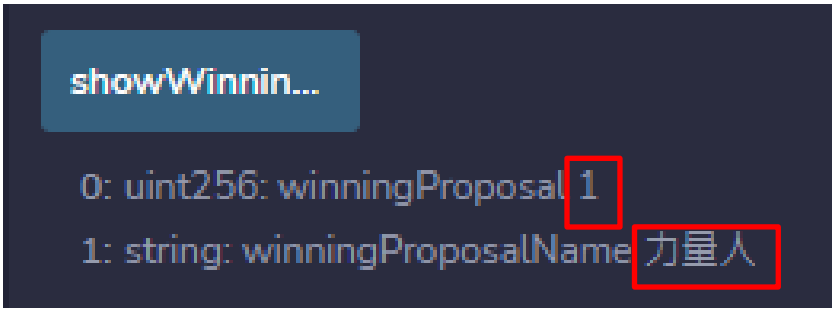


圖 15、開票結果

※補充:候選人資訊

本次選舉共兩位候選人。第 0 位候選人為”阿糖”，第 1 位候選人為”力量人”

```
//把所有候選人加進提案列表
proposals.push(Proposal({
    name: "阿糖",
    voteCount: 0
}));

proposals.push(Proposal({
    name: "力量人",
    voteCount: 0
}));
```

圖 16、候選人名單

備註:在一開始，因各種操作都需要支付一些 gas 費，所以在 3 個帳戶都有事先到 faucet 先拿一些測試 ETH 幣。

在這邊恭喜**力量人為最高票數**!

本專案有紀錄完整程式碼

完整程式碼發佈在 github 上:
https://github.com/LwGsTeemo/smart-contract-homework