Final Project Report

Project 1

合約地址: 0xd6df1978d26f378c0813b8f67ce7547083cf1da1

Logic

接受不同玩家的不同投注(大於entranceFee就好),並且玩家投注為累積的,可以重複投注, 勝率與該玩家總投注金額正比

Example:

entranceFee = 1

A先投3

B投5

C投2

A投2

結算時:

總金額 = 3+5+2+2=12

A總金額3+2=5

B總金額5

C總金額2

index = 亂數 % 12 = 贏家的money

index 落在0~4: A贏

index 落在5~9: B贏

index 落在10~11: C 贏

Structure and Member

An enum for status

```
1 enum lotteryStatus{
2  NO, // not playing
3  PLAYING, // playing
4  COUNTING // counting result
5 }
```

members

```
1 mapping(uint256=>address) private IDtoPlayer; // to emulate the traverse of players mapping
2 mapping(address=>uint256) private players; // 玩家到投注金額的mapping
3 uint256 public totalPlayer; // 全部參與的玩家
5 address public recentWinner; // 最新的赢家地址
6 uint256 public entranceFee; // 每次投注的threshold
7 address private _owner; // 合約持有者
9 uint256 private _totalMoney; // 總投注金額
10 lotteryStatus private _STATUS; // 現在遊戲的狀態
```

Modifier

To check if the function is execute by the owner

```
1 modifier isOwner(){
2    require(_owner == msg.sender);
3    _;
4 }
```

constructor

The constructor setting the onwer and status of the game to no

```
constructor() public {
    _owner = msg.sender;
    _STATUS = lotteryStatus.NO;
}
```

functions

```
function enter() public payable {
       // 檢查狀態是不是lotteryStatus.PLAYING跟輸入金額是否大於entranceFee
       require(_STATUS == lotteryStatus.PLAYING);
       require(msg.value >= entranceFee);
       // 把總金額+輸入金額
       _totalMoney += msg.value;
       // 如果玩家不存在, 則新增玩
       if(players[msq.sender] == 0)
11
           IDtoPlayer[totalPlayer++] = msq.sender;
12
13
       // 把金額送到該玩家的mapping
       players[msg.sender] += msg.value;
14
15 }
```

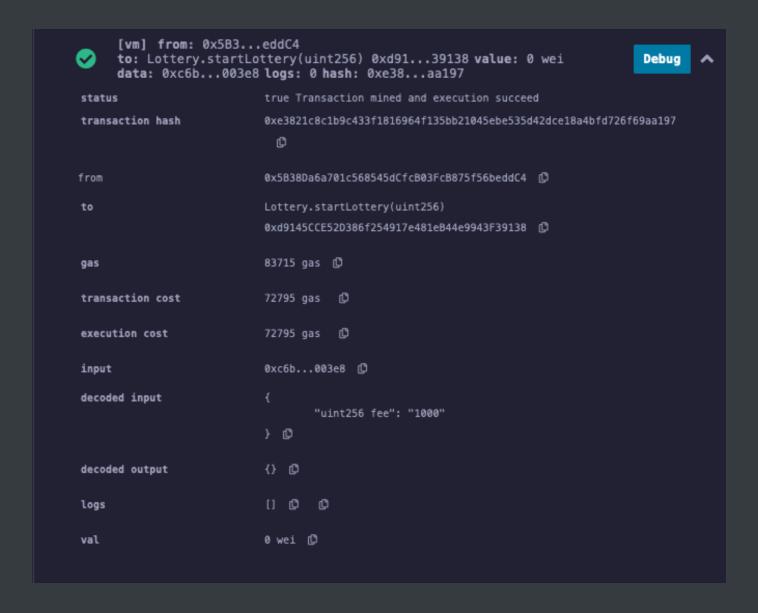
```
13// 改變狀態為正在遊玩14_STATUS = lotteryStatus.PLAYING;15}
```

```
1 // 檢查是否是owner執行
   function endLottery() public isOwner{
       // 檢查狀態是否是正在玩的狀態
       require(_STATUS == lotteryStatus.PLAYING);
       // 狀態設定為正在結算
       _STATUS = lotteryStatus.COUNTING;
       // 用亂數生成找出贏家的是介於哪個money區段的
10
       uint256 indexOfWinner =
   uint256(keccak256(abi.encodePacked(msg.sender, block.difficulty,
   block.timestamp))) % _totalMoney;
11
       // 目前累積金額跟贏家的index
12
       uint256 cur = 0;
13
14
       uint256 i:
15
17
       // 遍歷所有玩家並用IDtoPlayer找出現在的index是哪個address, 再用找到的
   address去找到目前金額
18
       // 如果大於目前金額代表這個index的money落在這個玩家的區間段,因此由這個玩家獲
       for(i = 0; i < totalPlayer; ++i){</pre>
19
           cur += players[IDtoPlayer[i]];
20
21
              if(cur >= indexOfWinner)
22
                  break;
23
24
       // 設定贏家
25
       recentWinner = IDtoPlayer[i];
26
27
28
       // 轉錢過去並確定是否成功
```

```
(bool finish, ) = payable(recentWinner).call{value: _totalMoney}
29
   ("");
       require(finish == true);
30
31
32
       // 把所有參賽者金額歸零(清空mapping)
33
       for(i = 0; i < totalPlayer; ++i)</pre>
34
         players[IDtoPlayer[i]] = 0;
35
36
       // 狀態設定為沒有在玩
37
       _STATUS = lotteryStatus.NO;
40 }
```

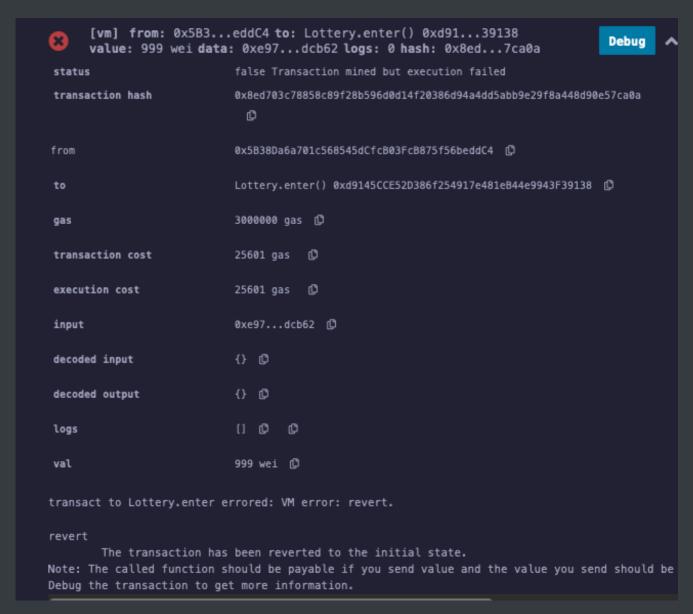
執行結果

startLottery:



enter:

1. 失敗(小於entreeFee)



2. 成功

	eddC4 to: Lottery.enter() 0xd9139138 ata: 0xe97dcb62 logs: 0 hash: 0x8f286c7b
status	true Transaction mined and execution succeed
transaction hash	0x8f221a779b993c34b91355c9c596affe612b75944a2a4ba95a76437462386c7b ☑
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 □
to	Lottery.enter() 0xd9145CCE52D386f254917e481eB44e9943F39138 🗓
gas	132306 gas 🗘
transaction cost	115048 gas 🗘
execution cost	115048 gas 🗓
input	მxe97dcb62 ტ
decoded input	↔ ©
decoded output	↔ ©
logs	
val	1001 wei 🗓

3. 成功(使用不同人投注):

<pre>Value: 1005 wei data: 0xe97dcb62 logs: 0 hash: 0xbadd6883</pre> Debug	
status	true Transaction mined and execution succeed
transaction hash	0xbad1c9c7a5dce3cdd73d67626984c9ff631f8537a64e7c559c42a4c6941d6883
from	0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
to	Lottery.enter() 0xd9145CCE52D386f254917e481eB44e9943F39138 🖟
gas	92976 gas 🗗
transaction cost	80848 gas 🗓
execution cost	80848 gas 🗘
input	0xe97dcb62 ₵
decoded input	() □
decoded output	O D
logs	
val	1005 wei ()

查詢入場費用:

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
                                                                                      Debug
      to: Lottery.entranceFee() data: 0x649...677e1
from
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 []
                            Lottery.entranceFee() 0xd9145CCE52D386f254917e481eB44e9943F39138
to
                            23604 gas (Cost only applies when called by a contract) (
execution cost
input
                            0x649...677e1 🖒
                            () ()
decoded input
decoded output
                                   "0": "uint256: 1000"
                            } 🖒

 ()

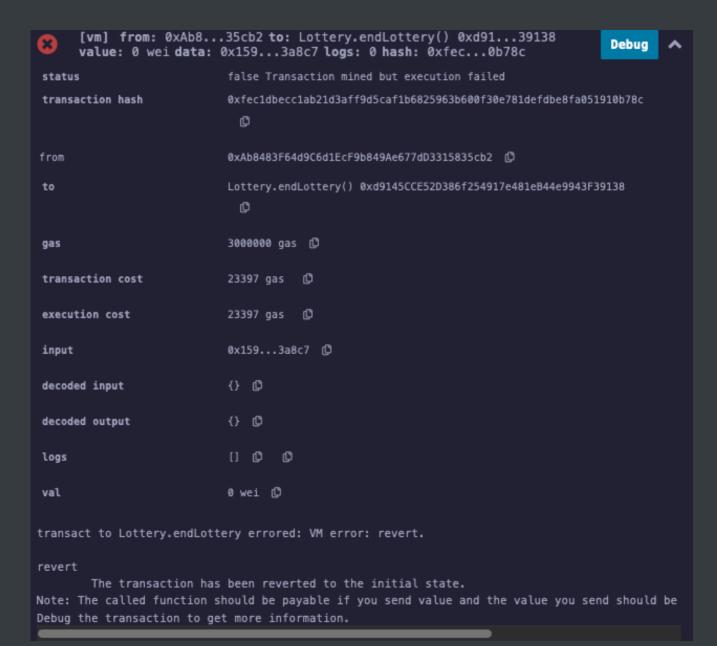
                                    Φ
logs
```

查詢入場人數:

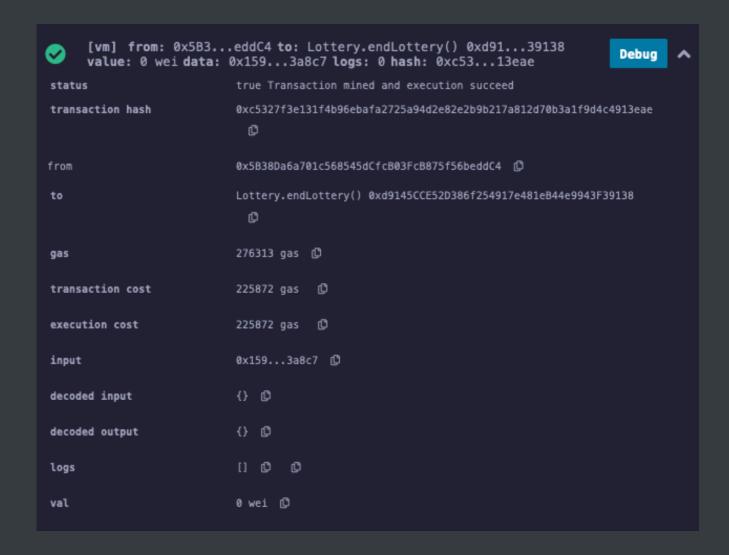
```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
                                                                                     Debug
     to: Lottery.totalPlayer() data: 0xc49...4a080
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 (
from
                            Lottery.totalPlayer() 0xd9145CCE52D386f254917e481eB44e9943F39138
                             Φ
execution cost
                           23537 gas (Cost only applies when called by a contract) 🚨
input
                           0xc49...4a080 🗓
decoded input
                            () Ø
decoded output
                                   "0": "uint256: 2"
                            } ©
                            [] 🗗
                                   Φ
logs
```

endLottery:

1. 失敗(非owner執行)



2. 成功



Project2

只列出與Project1不同的地方

Logic

- 1. 每次結算都記錄10%金額,在增加一個Function讓owner執行輸入提領的地址並進行提領
- 2. 提供publicMember查看
- 3. 提供publicMember查看
- 4. 提供publicMember查看
- 5. 新增一個struct存贏家address, 金額跟人數,然後用array包起來提供function輸入第幾輪來查詢
- 6. 新增一個mapping(慈善機構address=>每次提領的紀錄的陣列),新增一個function,給定地 址讓玩家輸入慈善機構address查詢前面提到的mapping

Structure and Member

```
1 // 紀錄每個round的狀態
2 struct roundStatus{
       address winnerAddress;
       uint256 winnerAmount;
       uint256 totalPlayer;
   RoundStatus[] private roundStatus; // 每個round的陣列
   mapping(address=>uint256[]) private charityRecord; // 提領紀錄
10
11
12
   uint256 public totalHistoryPlaytime; // 歷史總開獎次數
   uint256 public totalHistoryMoney; // 歷史總金額
13
14
   uint256 public totalHistoryPlayer; // 歷史總玩家
15
16 uint256 private _totalCharity; // 全部的慈善金額
```

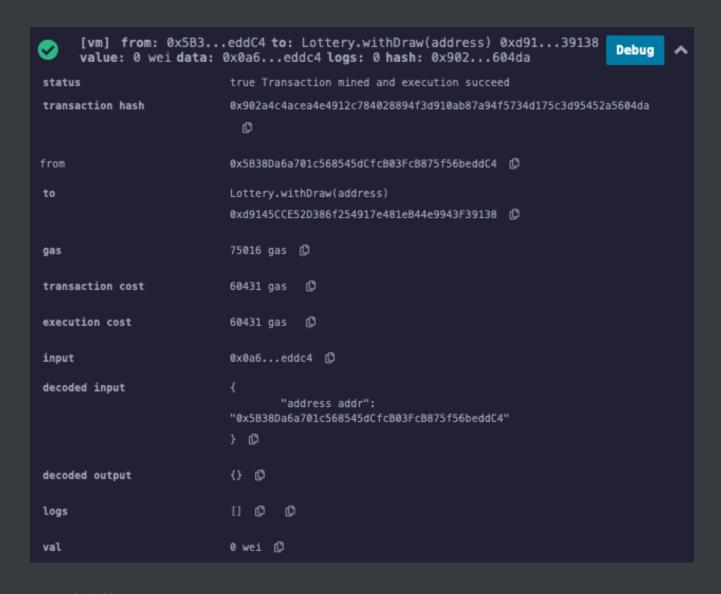
functions

```
15
       // 把提領紀錄到charityRecord
        charityRecord[addr].push(_totalCharity);
17
18
       // 歸零
       _totalCharity = 0;
19
20 }
21
22
    function endLottery() public isOwner{
23
24
25
26
       // 處理能查詢的變數
27
       ++totalHistoryPlaytime;
        totalHistoryMoney += _totalMoney;
28
29
        totalHistoryPlayer += totalPlayer;
30
31
       // 玩家拿9/10的總金額
        uint256 totalPay = _totalMoney * 9 / 10;
32
33
        (bool finish, ) = payable(recentWinner).call{value: totalPay}("");
        require(finish == true);
34
35
       // 把這輪狀態記下來
36
37
        roundStatus.push(RoundStatus(recentWinner, totalPay, totalPlayer));
38
       // 剩下的累積到慈善機構總額
39
       _totalCharity += _totalMoney - totalPay;
41 }
42
43
   // 查詢每輪狀態
    function roundQuery(uint256 index) public returns(RoundStatus memory){
44
45
        require(index < totalHistoryPlaytime);</pre>
46
        return roundStatus[index];
47 }
   // 杳詢提領紀錄
    function charityQuery(address addr) public returns(uint256[] memory){
50
```

```
return charityRecord[addr];
}
```

執行結果

withdraw(以自己):



歷史開獎次數:

```
call to Lottery.totalHistoryPlaytime
      [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
                                                                                      Debug
      to: Lottery.totalHistoryPlaytime() data: 0xef1...ffe22
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 ()
from
                            Lottery.totalHistoryPlaytime()
to
                            0xd9145CCE52D386f254917e481eB44e9943F39138 (C)
                            23603 gas (Cost only applies when called by a contract) (
execution cost
                            0xef1...ffe22 🗘
input
decoded input
                            {} @
decoded output
                                    "0": "uint256: 1"
                            } @
                            [] 🗗
                                    Φ
logs
```

歷史總金額:

```
call to Lottery.totalHistoryMoney
      [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
                                                                                     Debug
      to: Lottery.totalHistoryMoney() data: 0x17c...d222b
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 ()
from
                            Lottery.totalHistoryMoney()
to
                            0xd9145CCE52D386f254917e481eB44e9943F39138
execution cost
                            23538 gas (Cost only applies when called by a contract) 🗓
                            0x17c...d222b 🗓
input
decoded input
                            () ()
decoded output
                                   "0": "uint256: 2006"
                            ) D
logs
                            [] 🗗
                                    Φ
```

歷史總玩家:

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
                                                                                     Debug
     to: Lottery.totalHistoryPlayer() data: 0x873...6289c
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 ()
from
                            Lottery.totalHistoryPlayer()
to
                            0xd9145CCE52D386f254917e481eB44e9943F39138 ()
execution cost
                            23493 gas (Cost only applies when called by a contract) 🗯
input
                            0x873...6289c 🗓
                            {} D
decoded input
decoded output
                                   "0": "uint256: 2"
logs
                            [] 🗗
                                    Φ
```

查詢歷屆資訊:

```
[vm] from: 0x5B3...eddC4
      to: Lottery.roundQuery(uint256) 0xd91...39138 value: 0 wei
                                                                                       Debug
     data: 0x908...00000 logs: 0 hash: 0xa68...17678
status
                            true Transaction mined and execution succeed
transaction hash
                            0xa6810c2fdc064c1e380ac8b2496b8eeb778618cb038e7dad26b0dcb77bc17678
                              Φ
from
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 (C)
                            Lottery.roundQuery(uint256)
to
                            0xd9145CCE52D386f254917e481eB44e9943F39138 (
                            37963 gas 🗘
gas
transaction cost
                            33011 gas
                                        0
                            33011 gas
execution cost
                                       0
                            0x908...00000 🖒
input
decoded input
                                    "uint256 index": "0"
                            } 🗅
decoded output
                                    "0": "tuple(address,uint256,uint256):
                            0x5B38Da6a701c568545dCfcB03FcB875f56beddC4,1805,2"
                            } 🗅
                            [] O
                                     Φ
logs
```

查詢慈善機構提領紀錄(以自己):

