

合约部署报告

学号：16340293 姓名：张晓帆

智能合约的设计

基本结构体

- Joiner 结构体：主要是代表志愿者，其中有志愿者的地址，名字，公益时，以及参加的公益项目地址
- Publisher 结构体：主要代表发布者，其中有发布者的地址，名字，公益时，以及发布的公益项目
- Project 结构体：主要是项目名，项目的公益时间，项目的最多的参加人数，项目的给予公益时，项目发布者的地址，参加项目志愿者们的地址

```
struct Joiner{
    address _joinerAddress;
    string _joinerName;
    uint8 _allTime;
}

struct Publisher{
    address _publisherAddress;
    string _publisherName;
    uint8 _allTime;
}

struct Project{
    address _projectAddress;
    string _projectName;
    //项目时间
    uint8 _projectTime;
    //项目所需人数
    uint8 _maxCount;
    //项目公益时
    uint8 _maxJobTime;
    //项目结束标志
    bool _isEnd;

    address _publisher;
    address [] _joiners;
}
```

添加

- 添加志愿者，添加发布者很简单，将默认的参数传入就好，只是发布者创建时可以设置其拥有的公益时，志愿者默认的初始公益时为0；
- 添加项目就比较麻烦，需要首先确定项目的发布者是否拥有最多人数*最多公益时数的公益时，如果没有就不能创建，如果有就可以创建项目。同时在创建项目时默认的志愿者数为0，需设置志愿者的最大数量，志愿者参加公益的时间，获得的公益时。

参加

志愿者参加公益项目，需要满足当前项目志愿者没有达到最大志愿者限制。若当前项目志愿者的数量达到最大的数量，志愿者就不能参加该项目。

完成

公益项目完成后，由智能合约自动的从发布者的公益时向志愿者的公益时转账，默认的转移公益时数为项目设置的公益时数。

部署智能合约

使用Remix 调试部署

- 创建3个账户，账户3：部署合约，账户1：志愿者，账户2：发布者
- 添加志愿者：

The screenshot shows the Remix IDE interface. On the left, the 'Debug' console displays the transaction details for the 'addNewJoiner' function. The transaction was successful, with a hash of 0x52e980c3b568a254a31bb7b58e90768db09e9248b1da5553e7333543f535e0. The input data is a JSON object: {"string name": "147"}. On the right, the 'Plant at 0x54b...36746 (memory)' window shows the 'addNewJoiner' function being called with the name '147'. Below this, a list of functions is visible: addNewProject, addNewPublisher, finishProject, send, showJoiner, showProject, and showPublisher.

- 显示志愿者信息：

The screenshot shows the Remix IDE interface. On the left, the 'Debug' console displays the transaction details for the 'showJoiner' function. The transaction was successful, with a hash of 0x890b8e99f49105c91284e40b3e4a3c4939d95724b494777e7936be9d01d95a5f. The input data is a JSON object: {"address": "0x14723A09ACf6D2A600c2F7aA4AF308FDDC160C", "string name": "147", "projects": []}. On the right, the 'Plant at 0x54b...36746 (memory)' window shows the 'showJoiner' function being called with the address '0x14723A09ACf6D2A600c2F7aA4AF308FDDC160C' and the string 'name 147'. Below this, a list of functions is visible: finishProject, send, showJoiner, showProject, and showPublisher.

- 添加发布者

The screenshot shows the Remix IDE interface. On the left, the 'Debug' console displays the transaction details for the 'addNewPublisher' function. The transaction was successful, with a hash of 0x8754f72476695957a5975e76e7ce04face66408849d76a4a1c305525a4338bf. The input data is a JSON object: {"string name": "CA3", "uint8 times": 150}. On the right, the 'Plant at 0x54b...36746 (memory)' window shows the 'addNewPublisher' function being called with the name 'CA3' and the times '150'. Below this, a list of functions is visible: finishProject, send, showJoiner, showProject, and showPublisher.

- 显示发布者信息

[illegible][illegible]

- 添加项目

⌕


From:0xc3...a733c to:Plant.addNewProject(string,uint8,uint8,uint8,address) 0x54b...36746 value:0 wei data:0xb50...00000 logs:0 hash:0xc36...24045

Debug

status	0x1 Transaction mined and execution succeeded
transaction hash	0x3e6fab8011be46a3164c1afb8c03e53b97cc3974810b053fa91ba0ada24045
from	0xc36b7d915458eef540ada0608dfc2f44e8fa733c
to	Plant.addNewProject(string,uint8,uint8,uint8,address) 0x54be0edf518ae66602e48aa58b2ce28b3d036746
gas	30000000000 gas
transaction cost	437998 gas
execution cost	413974 gas
hash	0x3e6fab8011be46a3164c1afb8c03e53b97cc3974810b053fa91ba0ada24045
input	0xb50...00000
decoded input	<pre>{ "string name": "siCA", "uint8 projectTime": 8, "uint8 count": 2, "uint8 jobline": 8, "address publisher": "0xc36b7d915458eef540ada0608dfc2f44e8fa733c" }</pre>
decoded output	<pre>[]</pre>
logs	
value	0 wei

name:

"147"



transact

addNewProject

name:

"xICA"

projectTime:

8

count:


2

jobTime:

8

publisher:

0xca35b7d915458ef540ade6068dfe2f44e8fa733c



transact

addNewPublisher

name:

"CA3"

times:

150

- 志愿者加入项目

from:0x147...c160c to:Plant.AddInProject(address, address) 0x804...d9573 value:0 wei data:0xc09...a733c logs:0 hash:0x828...45fe9

Debug

status	0x1 Transaction mined and execution succeed
transaction hash	0x82801867edf31ddd39e9265c4ffda8fba3d966283c800adf0fee8fe95b5745fe9
from	0x14723a09acf0f624504cd7a54aff308f3dd160c
to	Plant.AddInProject(address, address) 0x8040095fb6806a9b19a4cd7b3cd96374dd9573
gas	3000000 gas
transaction cost	50270 gas
execution cost	26182 gas
hash	0x82801867edf31ddd39e9265c4ffda8fba3d966283c800adf0fee8fe95b5745fe9
input	0xc09...a733c
decoded input	{ "address joiner": "0x14723A09ACff6DCA60De27faA4AF308FD0C160C", "address project": "0xC35B7d915458BF540aB0606Dfe2744E8fa733c" }
decoded output	{ "0": "bool: isJoined true" }
logs	
value	0 wei

Transactions recorded: 5

Deployed Contracts

Plant at 0x804...d9573 (memory)

AddInProject

joiner: 0x14723a09acf6d2a60dcd7aa4aff308fddc160c

project: 0xca35b7d915458ef540ade6068dfe2f44e8fa733c

transact

addNewJoiner "147"

addNewProject

name: "xiCA"

projectTime: 8

[illegible]

```

0000000000000000,0x00000000000000000000000000000000,0xC3A5b7d915458EF
540aDe6068dFe2F44E8fa733c

showProject

0: address: a 0xC3A5b7d915458EF540aDe6068dFe2F44E8fa733c
1: string: name xICA
2: address: publisher 0xC3A5b7d915458EF540aDe6068dFe2F44E8fa733c
      3: address[]: joiner
0x0000000000000000000000000000000000000000000000000,0x00000000000000000000000000000000
00000000,0x14723A09ACf8D2A60Dcd7fA4A4F308FDDC160C

showPublisher

0: address: a 0xC3A5b7d915458EF540aDe6068dFe2F44E8fa733c
1: string: name CA3
      2: address[]: projects
0x0000000000000000000000000000000000000000000000000,0x00000000000000000000000000000000
00000000,0x0000000000000000000000000000000000000000000000000,0x00000000000000000000000000000000
000000000000000000,0x00000000000000000000000000000000000000000000000,0xC3A5b7d915458EF
540aDe6068dFe2F44E8fa733c

```

- 完成项目

Ethereum Wallet 账户 编辑 视图 开发 窗口 帮助

WALLETS SEND Private-net Connecting... 190 10s since last block CONTRACTS 950.00 ETHER*

Wallet Contracts

These contracts are stored on the blockchain and can hold and secure Ether. They can have multiple accounts as owners and keep a full log of all transactions.

+ ADD WALLET CONTRACT

Latest Transactions

Filter transactions

Nov 26	Created contract	Account 3 → Created contract at : PLANT 0D05	a few seconds ago	-0.00 ETHER	→
Nov 26	Transfer between accounts	Account 1 → Account 3	3 minutes ago	200.00 ETHER	↺
Nov 26	Transfer between accounts	Account 1 → Account 2	3 minutes ago	200.00 ETHER	↺
Nov 26	Sent	0x363EDFF9DE3F51a603d6bF3d780FB4024012074D → 0x62b165614565C43311f9589d5B0Fae74b9d93F32	2 hours ago	-200.00 ETHER	→
Nov 26	Sent	0x363EDFF9DE3F51a603d6bF3d780FB4024012074D → 0x73bf8c78Dde5e0db85b3ADdC02B50470956e6416	2 hours ago	-200.00 ETHER	→

Show More

• 添加志愿者

Ethereum Wallet 账户 编辑 视图 开发 窗口 帮助

WALLETS SEND Private-net Connecting... 190 55s since last block CONTRACTS 950.00 ETHER*

: PLANT 0D05 0.00 ETHER*

Read From Contract

Show joiner

A 0x00

Name

Times 0

Projects

Show publisher

A 0x00

Name

Times 0

Projects

Show project

A 0x00

Name

Publisher

Write To Contract

Select function

Add New Joiner

Name - string

“志愿者”

Execute from

Account 1 - 550.00 ETHER

SENDING...

Ethereum Wallet

账户 编辑 视图 开发 窗口 帮助

WALLETS

SEND

Private-net | Connecting... | 208 | 5s since last block

CONTRACTS

0.00 ETHER*

: PLANT 0D05

0.00 ETHER*

Times
0

Projects
0x00
0x00
0x00
0x00
0x00

Show publisher

A
 Account 2

Name
“发布者”

Times
150

Projects
0x00
0x00
0x00
0x00
0x00

Show project

A
 0x00

Name

Publisher
 0x00

Joiner

Times - 8 bits unsigned integer

150

Execute from
 Account 2 - 200.00 ETHER

EXECUTE

- 添加项目

Ethereum Wallet

账户 编辑 视图 开发 窗口 帮助

WALLETS

SEND

Private-net | Connecting... | 208 | 2 minutes since last block

CONTRACTS

0.00 ETHER*

: PLANT 0D05

HIDE CONTRACT INFO

0.00 ETHER*

Read From Contract

Show joiner

A
Account 1

Name
“志愿者”

Times
0

Projects
0x00
0x00
0x00
0x00
0x00

Show publisher

A
Account 2

Name
“发布者”

Times
150

Projects
0x00
0x00
0x00
0x00

Write To Contract

Select function

Add New Project

Name - string
“公益项目”

Project time - 8 bits unsigned integer
8

Count - 8 bits unsigned integer
2

Job time - 8 bits unsigned integer
8

Publisher - address
0xEC2ea1724cD820Cda0cD22268682817A8a589E19

Execute from
Account 2 - 200.00 ETHER

EXECUTE

Read From Contract

Show joiner

A

Name

“**志愿者**”

Times

8

Projects

```
0x00000000000000000000000000000000000000000000
```

```
0x0000000000000000000000000000000000000000000000000
```

```
0x00000000000000000000000000000000
```

```
0x000000000000000000000000000000000000000000000000000
```

```
0x000000000000000000000000000000000000000000000000000
```

Show publisher

A

Name

“发布者”

Times

126

Projects

```
0x00000000000000000000000000000000000000000000
```

```
0x000000000000000000000000000000000000000000000
```

```
0x000000000000000000000000000000000000000000
```

```
0x00000000000000000000000000000000
```

```
0x000000000000000000000000000000000000000000000000000
```

XX

遇到的困难和解决方法

一开始同时设计了4个合约，分别为志愿者合约，发布者合约，项目合约，平台合约。在remix上调试也成功了，但是部署到私链上时就会出现错误，一直以为是合约有问题，在翻看了控制台的日志文件后，发现是gas 限制量有点小，使用合约里的函数时需要的gas超过了限制。就重新设置了gaslimit 为0xffffffff000。但是还是有问题，调用的函数需要的gas 接近200以太币，gas太多，所以换为使用一个合约，将志愿者，发布者，项目 写成结构体，就减少了gas。可以正常部署了