

ZJU-blockchain-course-2022

↑ 可以修改成你自己的项目名。

第二次作业要求（可以删除）：

去中心化学生社团组织治理应用

- 每个学生初始可以拥有或领取一些通证积分（ERC20）。
- 每个学生可以在应用中可以：
 1. 使用一定数量通证积分，发起关于该社团进行活动或制定规则的提案（Proposal）。
 2. 提案发起后一定支出时间内，使用一定数量通证积分可以对提案进行投票（赞成或反对，限制投票次数），投票行为被记录到区块链上。
 3. 提案投票时间截止后，赞成数大于反对数的提案通过，提案发起者作为贡献者可以领取一定的积分奖励。
- (Bonus) 发起提案并通过3次的学生，可以领取社团颁发的纪念品（ERC721）

以下内容作为作业仓库的README.md中需要描述的内容。请根据自己的需要进行修改并提交。

作业提交方式为：提交视频文件和仓库的连接到指定邮箱。

如何运行

1. 在本地启动ganache应用（端口号8545）。
2. 在 `./contracts` 中安装需要的依赖，运行如下的命令：

```
1 | npm install
```

3. 在 `./contracts` 中编译合约，运行如下的命令：

```
1 | npx hardhat compile
```

4. 将后端代码部署在ganache链上，运行如下的命令：

```
1 | npx hardhat run scripts/deploy.ts --network ganache
```

5. 替换frontend\src\utils\abis中的MyERC20.json, MyERC721.json, StudentSocietyDAO.json文件为contracts\artifacts\contracts下的同名文件，在frontend\src\utils\contract-addresses.json中修改三个合约地址为中断打印出的对应地址。
6. 在 `./frontend` 中启动前端程序，运行如下的命令：

```
1 | npm run start
```

功能实现分析

简单描述：项目完成了要求的哪些功能？每个功能具体是如何实现的？

总体数据结构

```
1 struct Proposal {
2     uint32 index;           // index of this proposal
3     address proposer;      // who make this proposal
4     uint256 startTime;     // proposal start time
5     uint256 duration;      // proposal duration
6     string name;           // proposal name
7     string content;        // proposal content
8     uint32 voteNumber;      // the number of votor
9     uint32 agreement;      // the number of agreement
10    bool isend;
11    bool ispass;
12    mapping (address => bool) option;           // the option of each member
13    mapping (address => bool) voterList;        // the list of members who
    has voted
14 }
```

其中 `option` 字段没有实际使用，但进行了维护。后续可以拓展为记名投票。

具体功能

初始领取一些通证积分

- 该功能参考了彩票系统demo中的设计，在 `MyERC20` 中继承 `ERC20` 并添加一个新的方法 `airdrop`。通过 `claimedAirdropPlayerList` 记录已经领取的地址。限制每一个地址只能领取一次。

```
1 contract MyERC20 is ERC20 {
2
3     mapping(address => bool) claimedAirdropPlayerList;
4
5     constructor(string memory name, string memory symbol) ERC20(name,
symbol) {
6         _mint(msg.sender, 100000);
7     }
8
9     function airdrop() external {
10         require(claimedAirdropPlayerList[msg.sender] == false, "This user
has claimed airdrop already");
11         _mint(msg.sender, 10000);
12         claimedAirdropPlayerList[msg.sender] = true;
13     }
14 }
```

发起提案

后端新建一个 `Proposal` 结构体，将提案名称、内容、持续时间传入，发起者为 `msg.sender`，开始时间为 `block.timestamp`，其他字段采用默认初始值。

```
1     function newProposal(string memory name, string memory content, uint256
duration)public{
2         studentERC20.transferFrom(msg.sender, address(this),
PROPOSAL_AMOUNT);
```

```

3      proposalNumber++;
4      Proposal storage n = proposals[proposalNumber];
5      n.index = proposalNumber;
6      n.agreement = 0;
7      n.content = content;
8      n.duration = duration;
9      n.isend = false;
10     n.ispass = false;
11     n.name = name;
12     n.proposer = msg.sender;
13     n.startTime = block.timestamp;
14     n.voteNumber = 0;
15 }

```

前端采用 onNewProposal 函数进行调用，从输入框中获取提案名称、提案内容、持续时间作为参数传入。调用后更新当前状态数据，刷新页面。

```

1  const onNewProposal = async (title:string,content:string,duration:number) =>
2  {
3      if(account === '') {
4          alert('You have not connected wallet yet.')
5          return
6      }
7      if (StudentSocietyDAOContract && myERC20Contract) {
8          try{
9              await
10             myERC20Contract.methods.approve(StudentSocietyDAOContract.options.address,
11             proposalAmount).send({
12                 from: account
13             })
14             await
15             StudentSocietyDAOContract.methodsNewProposal(title,content,duration).send({
16                 from: account
17             })
18             const mc = await
19             StudentSocietyDAOContract.methods.getMemberCount(account).call()
20             setMemberCount(mc)
21             const pn = await
22             StudentSocietyDAOContract.methods.getProposalNumber().call()
23             setProposalNumber(pn)
24             const ind = await
25             StudentSocietyDAOContract.methods.getProposalID().call()
26             setDataIndex(ind)
27         }catch (error: any) {
28             alert(error.message)
29         }
30     }
31 }

```

对提案进行投票

后端主要采用 `vote` 函数进行投票，由前端传入投票的选择以及被投票提案的id，首先利用 `check` 函数进行状态的刷新。`msg.sender`将 `VOTE_AMOUNT` 浙大币转入本账户。对应提案的 `voteNumber` 加一，如果投票结果为true（赞同），则 `agreement` 加一。并将该用户的投票结果记录到对应 `Proposal` 的 `option` 中。

同时`check`函数也在前端与<查看最新状态>按钮相绑定。方便直接刷新。

```
1  function vote(bool option,uint32 proposalid)public{
2      check();
3      studentERC20.transferFrom(msg.sender, address(this), VOTE_AMOUNT);
4      Proposal storage _proposal = proposals[proposalid];
5
6      // only once
7      require(_proposal.voterList[msg.sender] == false, "This user has
voted already");
8      _proposal.voterList[msg.sender]=true;
9
10     require((_proposal.isend!=true) && (_proposal.startTime +
_proposal.duration > block.timestamp), "Voting has closed.");
11     if((_proposal.isend==false)) //the proposal is not over
12     {
13         _proposal.voteNumber++;
14         if(option == true)
15         {
16             _proposal.agreement++;
17         }
18         _proposal.option[msg.sender]=option;
19     }
20 }
```

提案投票时间截止后，赞成数大于反对数的提案通过

`check`函数：遍历当前所有的协议，对于 `isend` 为 `false` 并且 `startTime + duration <= block.timestamp` 的提案继续进行处理，将 `isend` 置为`true`，判断同意者是否超过全部投票者的半数，超过则将`ispass`置为`true`。

```
1  function check()public{
2      for(uint32 i = 1; i<=proposalNumber; i++)
3      {
4          Proposal storage _proposal = proposals[i];
5          // is not end
6          if((_proposal.isend==false)&&(_proposal.startTime +
_proposal.duration <= block.timestamp))
7          {
8              _proposal.isend = true;
9              if(_proposal.agreement > (_proposal.voteNumber/2)) //
pass
10              {
11                  _proposal.ispass = true;
12                  uint32 count = 2*_proposal.voteNumber-1; //
voteNumber != 0
```

```

13         studentERC20.transfer(_proposal.proposer,
PROPOSAL_AMOUNT+ count*VOTE_AMOUNT);
14         memberCount[_proposal.proposer] ++;
15         if(memberCount[_proposal.proposer]%3==0)
16         {
17             myERC721.bonus(_proposal.proposer);
18         }
19     }
20 }
21 }
22 }

```

提案发起者在提案通过后领取一定的积分奖励

当check函数判断其为通过后，向提案发起者转账 `studentERC20.transfer(_proposal.proposer, PROPOSAL_AMOUNT+ count*VOTE_AMOUNT);` 其中 `count` 为 `2*voteNumber-1`，即投票者越多，该提案越有价值，通过时发起者奖励越多。

- check函数见上

发起提案并通过3次的学生，可以领取社团颁发的纪念品（ERC721）

```

1 mapping(address=>uint32) memberCount;

```

check函数中，当判断有一个提案通过时，对应用户提案通过数加

一：`memberCount[_proposal.proposer] ++;`，每当 `memberCount[_proposal.proposer]%3==0` 时，奖励其一个纪念品。（check函数见上）

同时在MyERC721中写bonus函数，用于发放奖励。

```

1 contract MyERC721 is ERC721 {
2     uint32 bn = 0;
3     constructor(string memory name, string memory symbol) ERC721(name,
symbol) {
4     }
5
6     function bonus(address to) external {
7         _mint(to, bn);
8         bn++;
9     }
10 }

```

项目运行截图

放一些项目运行截图。

项目运行成功的关键页面和流程截图。主要包括操作流程以及和区块链交互的截图。

初始页面



连接钱包并领取空投



成功领取空投，此时该账户有浙大币10000



发起新提案

社团提案管理页面

领取浙大币空投

查看最新状态

当前用户: 0xe992e203675c4a70c3d9ac03d442f59066546a16
当前用户拥有浙大币数量: 10000 通过提案数量: 0 纪念品数量: 0

开会

时间: 11.6
地点: 浙江大学玉泉校区

120

花费500发起提案

已有0个提案

	总投票数	赞同数	是否结束	是否通过
No Data				

Localhost 8545

hw1

检测到新地址! 点击添加至地址簿。

详情 数据 十六进制文件

编辑

估计燃料费用 0.00641798
0.006418 ZJUTOKEN

推荐站点 最高收费: 0.00641798 ZJUTOKEN

总额 0.00641798
0.00641798 ZJUTOKEN

金额 + 燃料费 最大金额: 0.00641798 ZJUTOKEN

拒绝 确认

此时发起了新提案

社团提案管理页面

领取浙大币空投

查看最新状态

当前用户: 0xe992e203675c4a70c3d9ac03d442f59066546a16
当前用户拥有浙大币数量: 10000 通过提案数量: 0 纪念品数量: 0

开会

时间: 11.6
地点: 浙江大学玉泉校区

120

花费500发起提案

已有1个提案

提案名	开始时间	结束时间	发起者	总投票数	赞同数	是否结束	是否通过	Action	Action
+ 开会	2022-11-06 16:16:04	2022-11-06 16:18:04	0xe992e203675c4a70c3d9ac03d442f59066546a16	0	0	未结束	未结束	同意	不同意

通过提案前面的 + 号可以查看具体内容

	提案名	开始时间	结束时间	发起者
	开会	2022-11-06 16:16:04	2022-11-06 16:18:04	0xe992e203675c4a70c3d9, 66546A16

时间: 11.6 地点: 浙江大学玉泉校区

用同一个账户进行投票，此处点击同意



成功花费50进行投票



等过了投票时间后查看最新状态，提案的状态刷新。该用户通过提案数变为1，浙大币增加 $500 + 50 * (2 * 1 - 1)$ ，变回10000。



该账号再次发布几个提案，对于最后一个名为aa的提案，用本账号与另外2个账号投2：1票

← → ↻ localhost:3000/#

社团提案管理页面

领取浙大币空投

查看最新状态

当前用户: 0xe992e203675C4A70C3d9Ac03D442F59066546A16

当前用户拥有浙大币数量: 9650 通过提案数量: 3 纪念品数量: 1

提案名

提案内容

有效时间(s)

花费500发起提案

已有4个提案

	提案名	开始时间	结束时间	发起者	总投票数	赞同数	是否结束	是否通过	Action	Action
+	开会	2022-11-06 16:16:04	2022-11-06 16:18:04	0xe992e203675C4A70C3d9Ac03D442F59066546A16	1	1	已结束	通过	同意	不同意
+	123	2022-11-06 16:21:06	2022-11-06 16:21:28	0xe992e203675C4A70C3d9Ac03D442F59066546A16	1	0	已结束	不通过	同意	不同意
+	123	2022-11-06 16:21:31	2022-11-06 16:21:53	0xe992e203675C4A70C3d9Ac03D442F59066546A16	1	1	已结束	通过	同意	不同意
+	aa	2022-11-06 16:22:09	2022-11-06 16:24:09	0xe992e203675C4A70C3d9Ac03D442F59066546A16	3	2	已结束	通过	同意	不同意

可以看到所有提案都正确执行，0xe99...账户有三个通过的提案，获得了一个ERC721的纪念品。

此时再次点击领取空投，右下角与上方均有报错

localhost:3000 显示

您已经领取过

确定

提案名

提案内容

有效时间(s)

花费500发起提案

已有4个提案



	总投票数	赞同数	是否结束	是否通过	Action	Action
0xe992e203675C4A70C3d9Ac03D442F59066546A16	1	1	已结束	通过	同意	不同意
0xe992e203675C4A70C3d9Ac03D442F59066546A16	1	0	已结束	不通过	同意	不同意
0xe992e203675C4A70C3d9Ac03D442F59066546A16	1	1	已结束	通过		
0xe992e203675C4A70C3d9Ac03D442F59066546A16	3	2	已结束	通过		

Google Chrome

Failed transaction

Transaction 451 failed! [ethjs-query] while formatting outputs from RPC '{"value": {"code": -32603, "data": {"message": "v

重复投票或结束后投票叶会报错

通过	同意	不同意
不通过	同意	不同意
通过	<div><div> Google Chrome</div><div><div>Failed transaction Transaction 455 failed! [ethjs-query] while formatting outputs from RPC '{"value": {"code": -32603, "data": {"message": "\v</div></div></div>	

参考内容

课程的参考Demo见：[DEMOS](#)。

如果有其它参考的内容，也请在这里陈列。