

区块链版狼人杀文档

区块链版狼人杀文档	1
选题背景	1
选题构思	2
部署合约及获取实例	3
使用说明	5
测试	15
Github 网址	28

选题背景

狼人杀是坊间非常流行的一种桌游，其游戏规则大致如下：

必须由一个熟悉游戏规则的人做法官。法官只负责游戏的进行，并不参与游戏，亦不属于狼人或村人。

法官根据玩家人数以及使用的身份牌的情况，拿出一定数量的身份牌，牌面向下，分发给各个玩家，每人一张。每个玩家看了自己的身份牌后，将身份牌牌面向下置于自己面前。

1 法官宣布“天黑请闭眼”，此时所有玩家都闭眼，进入天黑（夜晚）阶段。随后法官宣布“情侣请闭眼”。

2 法官宣布“盗贼请睁眼”，盗贼可以交换身份或者不交换。结束后，法官宣布“盗贼请闭眼”。

3 法官宣布“丘比特请睁眼”，丘比特睁眼并指定两名玩家成为情侣。随后法官宣布“丘比特请闭眼”，丘比特闭上眼睛。法官绕场一周，悄悄拍两名情侣的头。

4 法官宣布“情侣请睁眼”，此时两名情侣睁眼，互相认识，但并不知道对方身份。接着法官宣布“情侣请闭眼”。

以上 2-4 步只有在第一个夜晚才会出现。即使没人抽到盗贼或丘比特，也要假装进行上面相应的步骤。

5 法官宣布“守卫请睁眼”，问守卫本轮会守护哪个人。（守卫不知道他保护的人的身份，或是否是被杀死的。）

6 法官宣布“预言家请睁眼”，预言家睁眼指定一名玩家，法官把该玩家的身份牌给预言家看，看完后身份牌放回原处，牌面向下。随后法官宣布“预言家请闭眼”。

7 法官宣布“狼人睁眼”，狼人睁眼相互确认，法官宣布“狼人开始杀人”，狼人一起指定一名玩家，该玩家在天亮时会死去。此时小女孩可以偷看。随后法官宣布“狼人请闭眼”。

8 法官宣布“女巫睁眼”，法官用手势告诉女巫刚才狼人杀死的是谁。女巫可以使用药剂，也可以不使用。如果女巫要使用药剂，则拇指向上表示用解药，救刚才被狼人杀死的人；拇指向下表示使用毒药，并且用手势告诉法官，要在哪位玩家身上使用毒药。被使用了毒药的玩家，天亮时死去。随后法官宣布“女巫闭眼”。

9 法官宣布“天亮了，所有人睁眼”。此时进入白天阶段。玩家睁眼后，法官宣布昨晚死去的人是谁（也可能没有人死去）。如果猎人被杀，则猎人立即进行报复，指定一名玩家，该玩家立即死去。如果爱人中的一个被杀，则另一名爱人立即死去。死去的玩家，需要发动技能的则翻开身份牌，没有技能可发动的不得翻开身份牌，之后出局，退出游戏，此后不得与其他玩家有任何交流。

10 场上活着的玩家按顺序依次进行讨论，决定白天要处死谁。（变体规则：刚才天黑时被杀的玩家左手边的玩家开始，按照顺时针方向，轮流发言，玩家之间不得对话，一名玩家发言时，其余玩家不得说话。）

11 讨论结束，开始投票。所有玩家都伸出手，法官一声令下，玩家同时把手指向自己心目中要处死的目标，被指得最多的玩家（警长的 1 票算作 1.5 票）被投票出局。出局玩家翻开自己的身份牌，退出游戏，此后不得与其他玩家有任何交流。如果投票出现平局，则没有人出局。（变体规则：平局时，得票数相同的玩家要再次发言，轮流发言一次，随后便再次投票，直至有人出局。）

12 法官宣布“天黑请闭眼”，所有人闭上眼睛。接着游戏跳回到前面的第 4 步（先知睁眼阶段），并按照这个顺序循环往复进行，直到游戏结束。

从规则中可以看出：

- 狼人杀是一款去中心化的游戏
- 且其规则较为复杂，为智能合约的执行提供了前提
- 并且在白天/黑夜的周期之内执行不同的行为，类似于区块产生的周期性
- 投票投死某人的行为可以类比为在账本上记账
- 每个人的发言或煽动是为了在玩家间达成共识
- 每个人的发言和每一晚的结局都将会向所有人广播
- 狼人杀依靠其他玩家以往的发言进行推理，要求其具有可追溯性和不可篡改性

可以看出，狼人杀的游戏规则与区块链的运行逻辑有较高的相似性，因此将区块链与狼人杀结合天然具有较高的可行性。

选题构思

在以太坊上搭建一条链，模拟进行狼人杀游戏的全过程，达到以下几点目标：

- 链上的各个节点对应不同的玩家
- 投票结果类似于记账
- 发言和投票需要付出代币
- 最终获得的玩家会获得代币
- 实现匿名性
- 由系统担任法官，广播投票结果和角色的生死
- 将复杂的规则作为智能合约与链绑定
- 可以看到所有人曾经的发言
- 人的发言是不可篡改的

部署合约及获取实例

1. ubuntu 建立连接

参考链接：

https://blog.csdn.net/qq_36303862/article/details/84405030?from=singlemessage 及 https://blog.csdn.net/loy_184548/article/details/79517264

在 Linux 下输入如下命令：

```
sudo apt-get install nodejs
sudo apt-get install npm
mkdir web3test && cd web3test
npm init
npm install web3@0.20.0 --save
```

开启服务：

```
cd ethereum/
geth --port 3000 --networkid 58343 --nodiscover --datadir=./private --
maxpeers=0 --rpc --rpcport 8543 --rpcaddr 127.0.0.1 --rpccorsdomain "*" -
-rpcapi "eth,net,web3,personal,miner"
```

```
geth attach http://127.0.0.1:8543
```

验证连接成功:

```
wjn@DESKTOP-83GQ0Q7:~/web3test$ nodejs
> var Web3 = require("web3");
undefined
> var web3 = new Web3();
undefined

> web3.setProvider(new Web3.providers.HttpProvider("http://localhost:8543"));
undefined

> web3.isConnected()
true
> web3.eth.accounts[0]
'0x9a31a67a49c9b406fbfa51e845887c5c33f12ce6'

Contract mined! address: 0x08f466786173f925d5a03c5724185c8407c69ee2 transactionHash: 0x60d68bd7487caa6a6ecf38df5bd8498f59cf611bf5808a0a7bfd8dd43cd21480

> killwolf.start_game.sendTransaction({from: account0, gas: 5000000});
'0xbceff55ef407bd9e5285eb86aac14099e527cf6a20b1e4939091ff7c3f2c5072'

> killwolf.get_counts_wolf.call().toNumber()
1
> killwolf.get_counts_man.call().toNumber()
3
```

2. 考虑到 Linux 上调试前端较为不便, 于是又使用 truffle 框架在 windows 下布局。

参考链接: <http://blog.hubwiz.com/2018/06/07/ethbox-readme/>

初始化文件夹内容

```
truffle install
```

开启模拟器

```
ganache-cli
```

编译合约

```
truffle.cmd compile  
truffle.cmd migrate
```

在 app.js 中自动获取 abi 和 address 来得到合约的实例：

```
var express = require('express')  
var fs = require('fs')  
var app = express()  
var Web3 = require("web3")  
var web3 = new Web3()  
var daystate = "day"  
var online = [0, 0, 0, 0, 0, 0, 0, 0]  
web3.setProvider(new Web3.providers.HttpProvider("http://localhost:8545"))  
var killwolf  
fs.readFile('./build/contracts/KillWolf.json', 'utf-8', function(err, data)  
{  
  if (err) {  
    console.log('文件读取失败');  
  } else {  
    //从本地读取编译结果以动态创建合约  
    //console.log(JSON.parse(data).abi);  
    var abi = JSON.parse(data)["abi"]  
    var address = JSON.parse(data).networks  
    var tl = Object.keys(address).length-1  
    address = address[Object.keys(address)[tl]].address  
    //console.log(address)  
    killwolf = web3.eth.contract(abi).at(address)  
    ...  
  }  
}
```

使用说明

- 安装 ethbox 环境

同时安装 express 作为后台框架。

- 开启仿真器

```

C:\Users\13371>ganache-cli
Ganache CLI v6.1.0 (ganache-core: 2.1.0)
>
Available Accounts
=====
(0) 0xbff709e5c2c236b17fc556841b4280b0496759e8
(1) 0x4e83d04f7afe3e6e41c22495e667c5f2a06bbf70
(2) 0x437502d2d3bda899cc6c2b89625cd8941fa9ae7c
(3) 0xf99c42f92198152ae21050a7870db8acb5b073c9
(4) 0x370926c9fef985a307e175eae8281f005f74617b
(5) 0x931a5b10da1521a59904aeaf19b3cfd7dd77442
(6) 0x0ee91b9fea49438d074afc18838cf8afbfa2fea2
(7) 0x51d4678b939dfd56139b0a4d0eee8d42b035b2fc
(8) 0x7a603c97ffdaa01c8d34791f520a5421a1469c4a
(9) 0x7750b8d93a88dbfd023716fe3fed0ecf29f65d10

Private Keys
=====
(0) 4865b80c94b82f588d338319b82704b6f07ab9cac88382e943e83caa693b5282
(1) af88f1673bc91341915a48197f788402c9d8435a684ebcf38c295ed20ca4929f
(2) d528ee81aae63c7087b89ba9358035638e16ba431811562f3f7a7a63bc78c9a8
(3) 34605ac1c648c5d6fb6db3eb7adaff1ca346e5dd47d3fa442f40617cb719c7e4
(4) 71a0fe9023ab3a8ffd7a6e43c0b5fa14adb8b130bc7494e8b86746c7deae955a
(5) 6d03cb07976dc124fe4c621de2a622cclc88f1075157f87efd292c4623bebb51
(6) 674d356b598ce23044ee72a340dd5e45cd84d7f286be7fb0c06738ebe2c92e44
(7) b2009573977c78e6fe9a94f0c18dalaf9af9a570ccc0bc2265364c8cc5d134fa
(8) 1357a029daa0beeb4ce385a7562bb55fee8142374876b850fc69e896d52c588d
(9) 5c645b4b5edb718f0b657d9c2036a180aaeb052d3c0b3d6f689b37f11e908503

HD Wallet
=====
Mnemonic:      blue inmate spy expect alarm layer bird guide number walk usage capable
Base HD Path:  m/44'/60'/0'/0/{account_index}

Listening on localhost:8545

```

- 编译部署合约

```

C:\Users\13371>cd killwolf
C:\Users\13371\killwolf>truffle.cmd compile
C:\Users\13371\killwolf>truffle.cmd migrate
Using network 'development'.

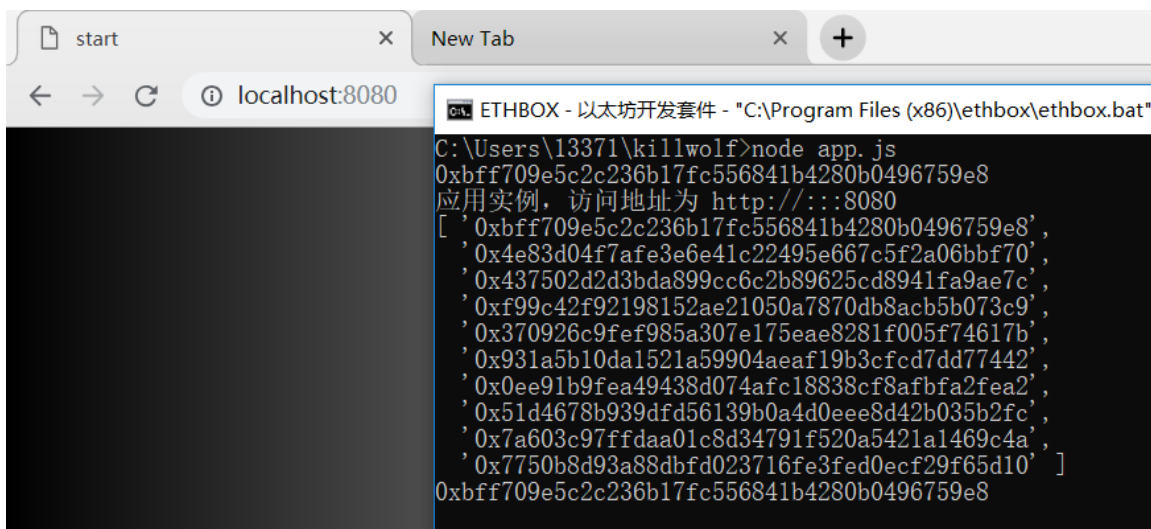
Running migration: 1_initial_migration.js
  Deploying Migrations...
... 0xf28e18c74b873f097ee4f3f2719b2168cf14650023e2d5d64871b6ce4b0cdf09
  Migrations: 0x11309386c29e09926b24b2c4b3b595fe7e420b57
Saving successful migration to network...
... 0x674326478283d70c36b244c2be9da99632f9b4943d3df80d75af08f4c6f2f877
Saving artifacts...
Running migration: 2_initial_killwolf.js
  Deploying KillWolf...
... 0x1bdee56ef93fe4f28b817ef388a0771b8c64085e11b752660d60d666e6246203
  KillWolf: 0x4d2d430alacbab7814140cada635eae229a9df
Saving successful migration to network...
... 0xa5dadcd0c1d6f2857764e9442c1dfd59c83cebeacdb44598d4e037878f3ee7ff
Saving artifacts...

```

- 运行 app.js(第一行为调用函数打印合约中法官的地址)

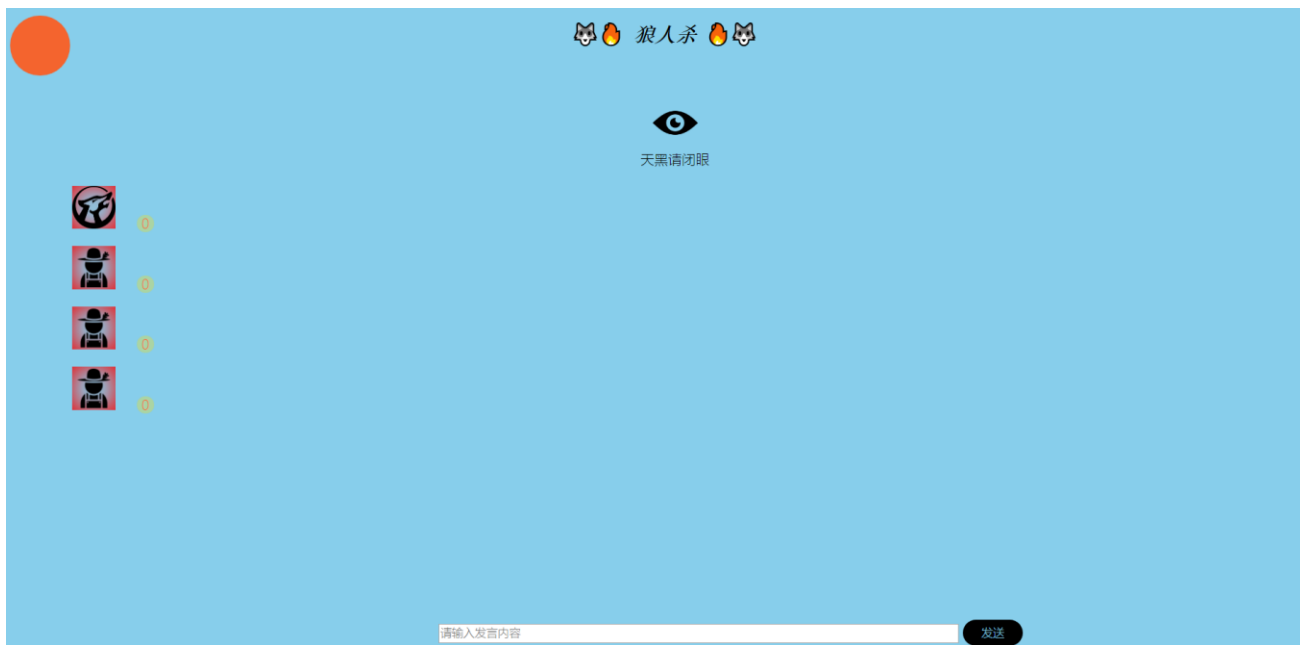
```
C:\Users\13371\killwolf>node app.js
0xbff709e5c2c236b17fc556841b4280b0496759e8
应用实例，访问地址为 http://:::8080
```

- 在浏览器中输入网址 `localhost:8080`，此时会出现开始游戏的界面，同时命令行中会显示链上的所有可用账号以及法官的地址。

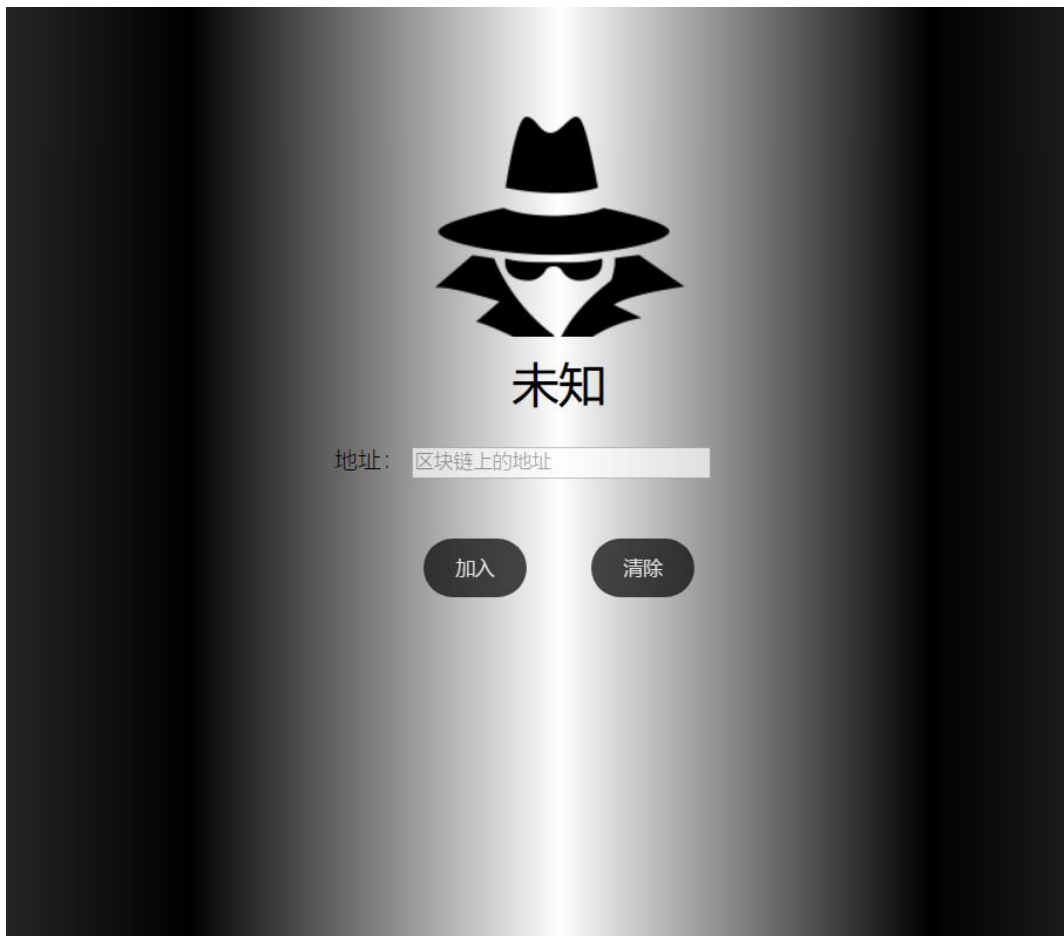


- 输入正确的地址后点击开局开启游戏（图标红色表示尚未登录可以看到所有人）





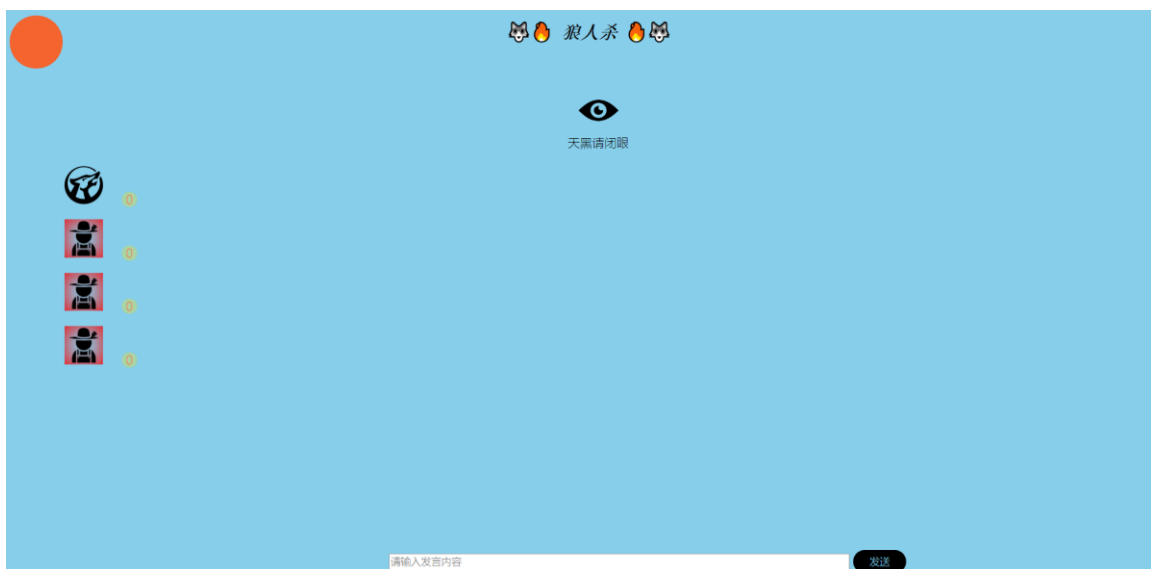
- 此时再次打开 localhost:8080 会进入游戏登录界面



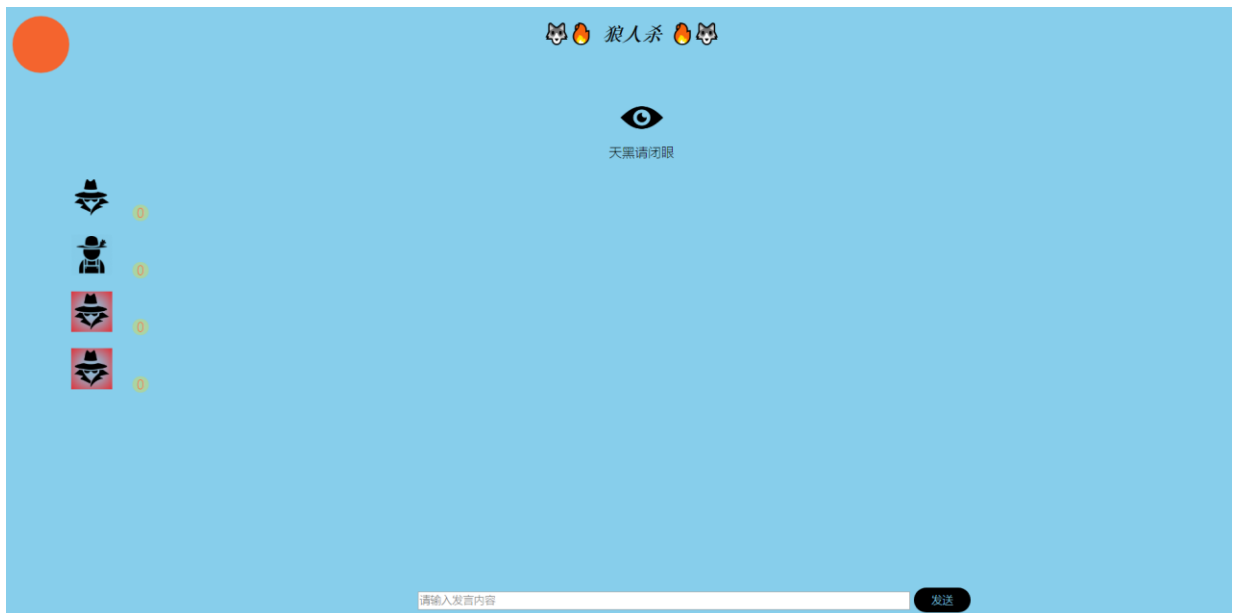
- 输入地址后会根据合约返回信息修改头像和身份。



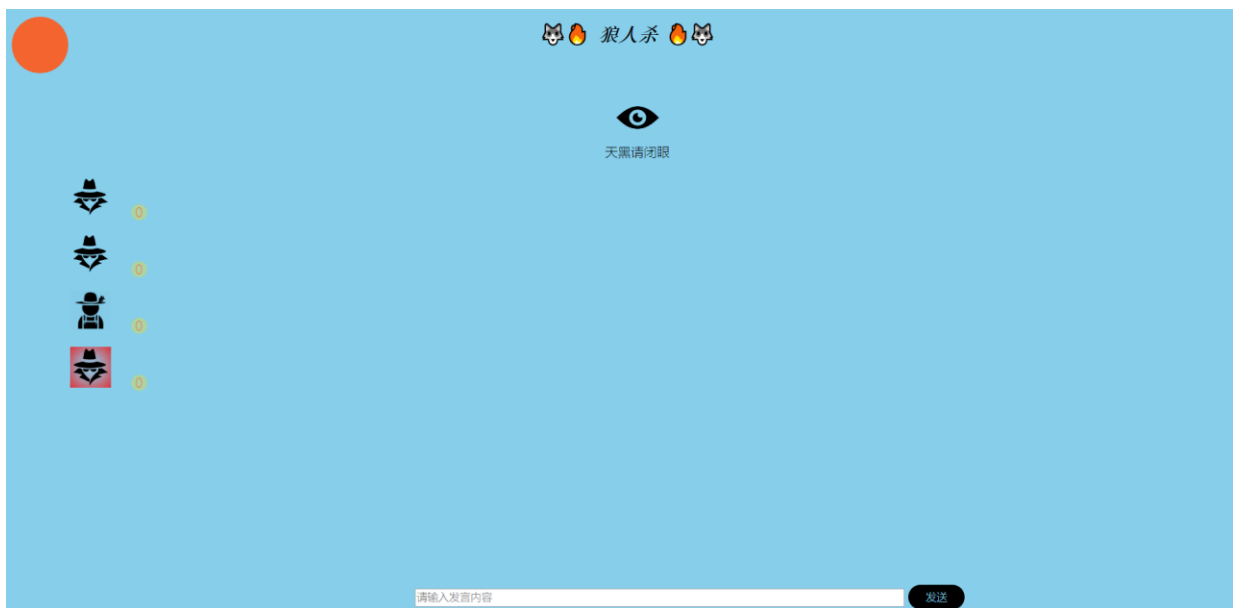
- 点击加入，狼人可以看到所有人的头像，自己的头像变为登陆状态。



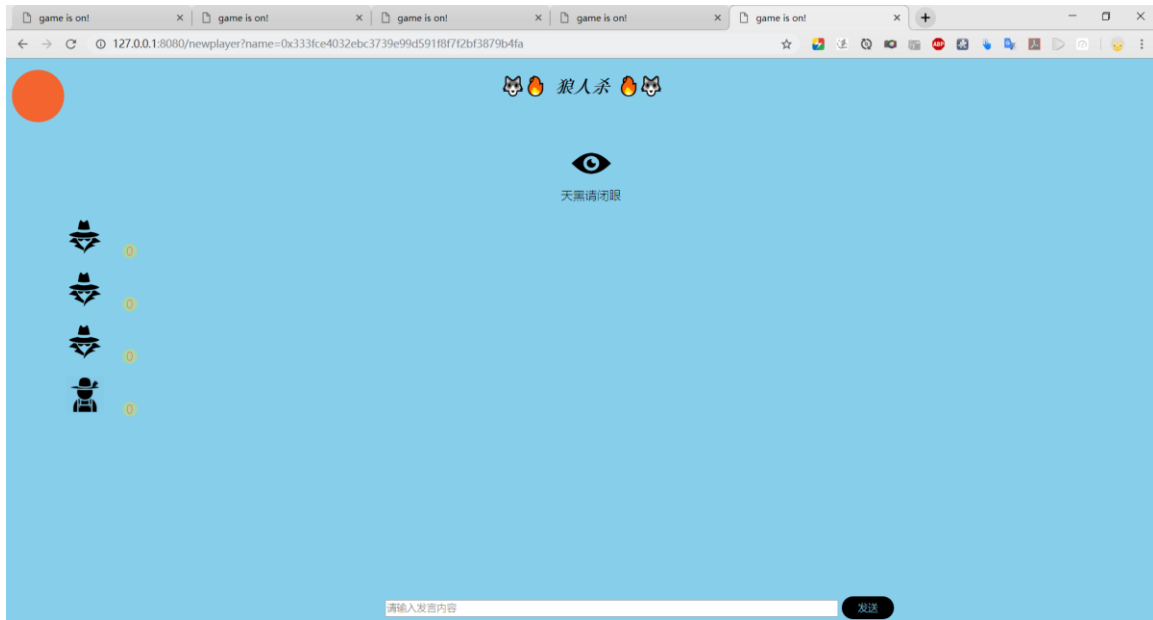
- 村民登陆游戏，村民只能看到自己。



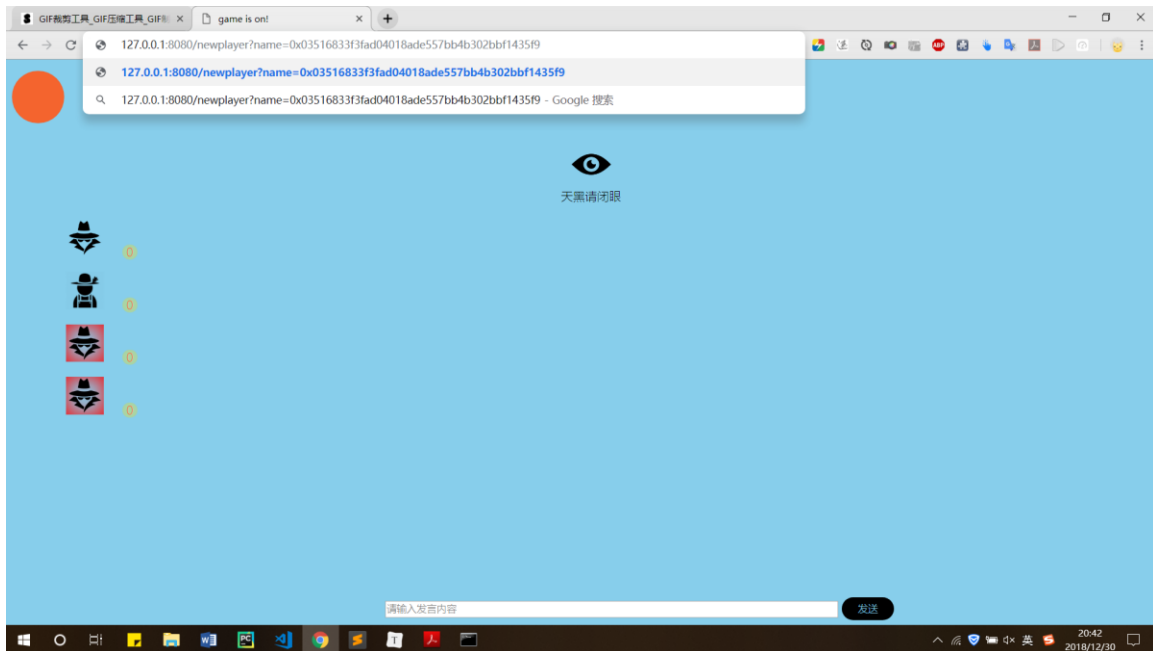
- 村民依次登陆。



- 法官和四个玩家登陆，由于 **Chrome** 浏览器对 **socket** 的并发数量的限制，每打开一个网页进行完操作后就需要关闭，再打开下一个。



- 直接在 url 中修改 name 后面的地址也是可以跳转到新的页面的



- 夜间点击头像投票



命令行也会在将投票写入合约后做出相应的提示。

```
vote completed!
```

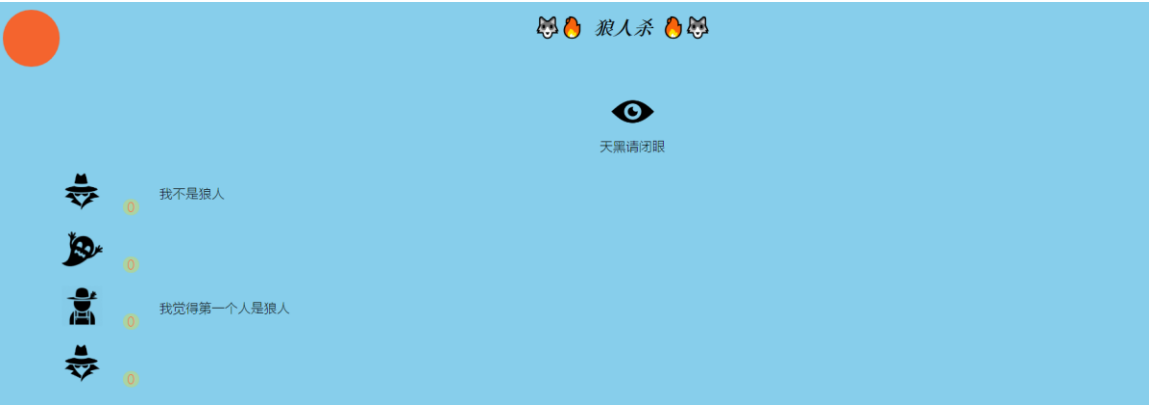
- 关闭狼人界面，以村民身份登入，投票数已从链上更新



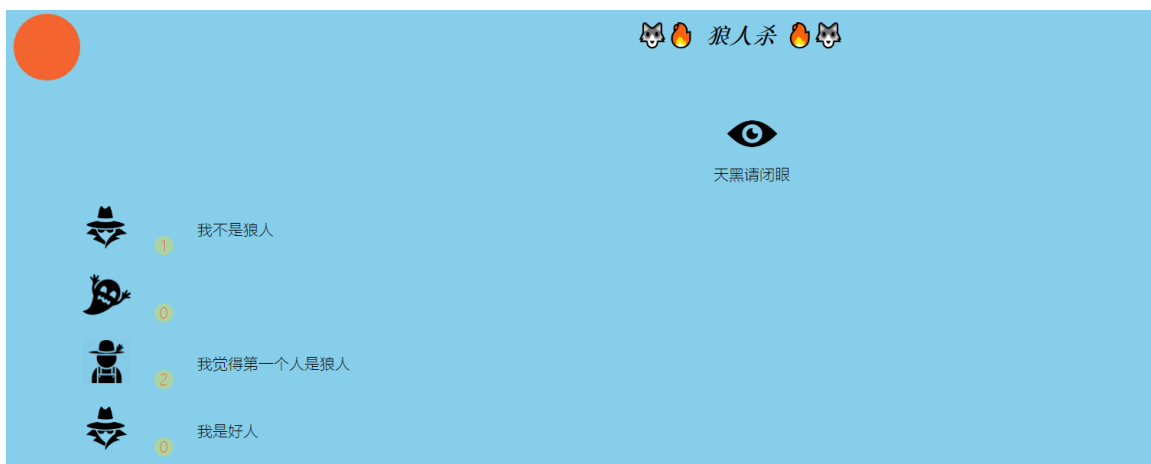
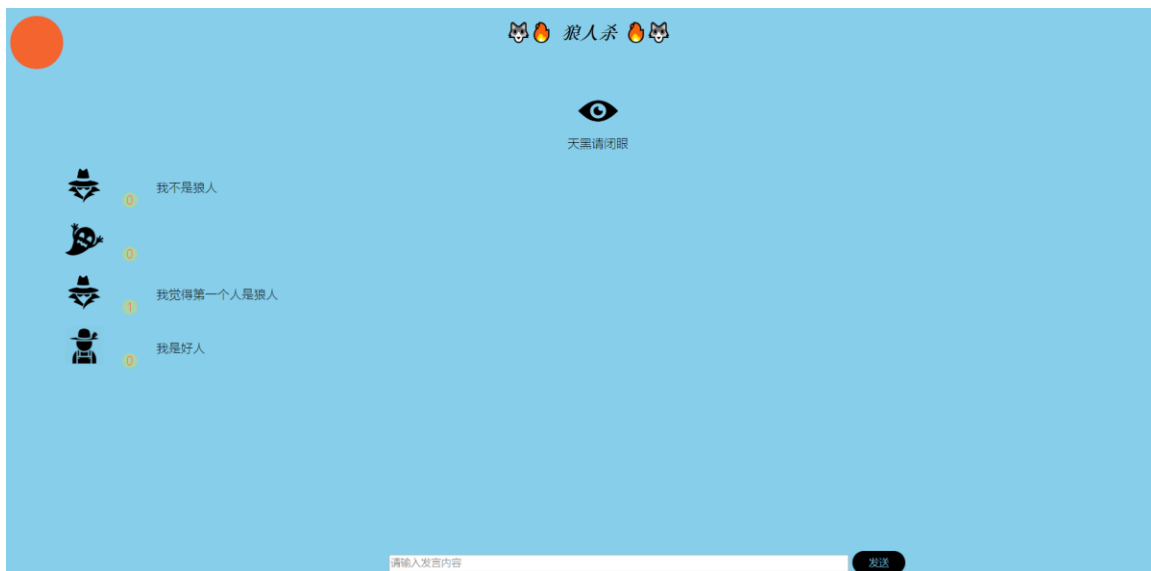
- 以法官身份登入，点击天亮请睁眼，夜晚被投死的人头像变成了鬼。



- 活着的人依次发言

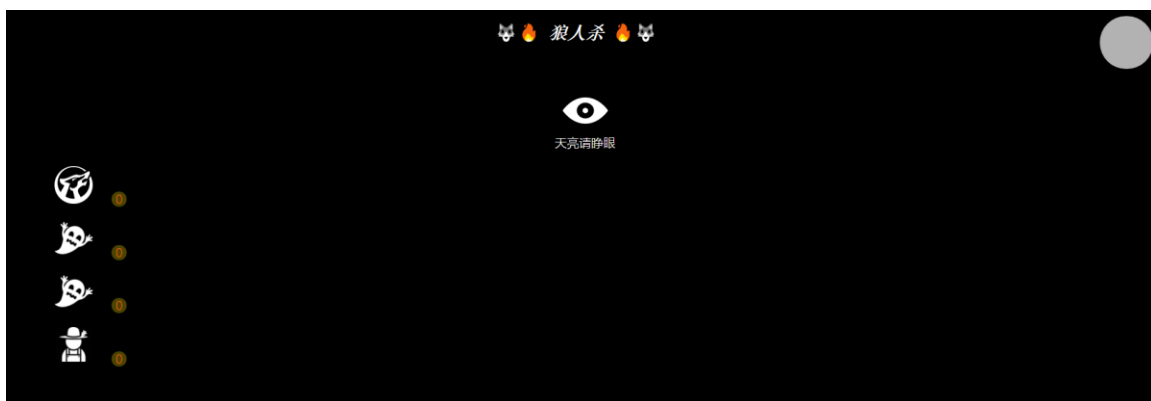


• 所有人发言完毕后进入白天投票环节，所有人必须投票。

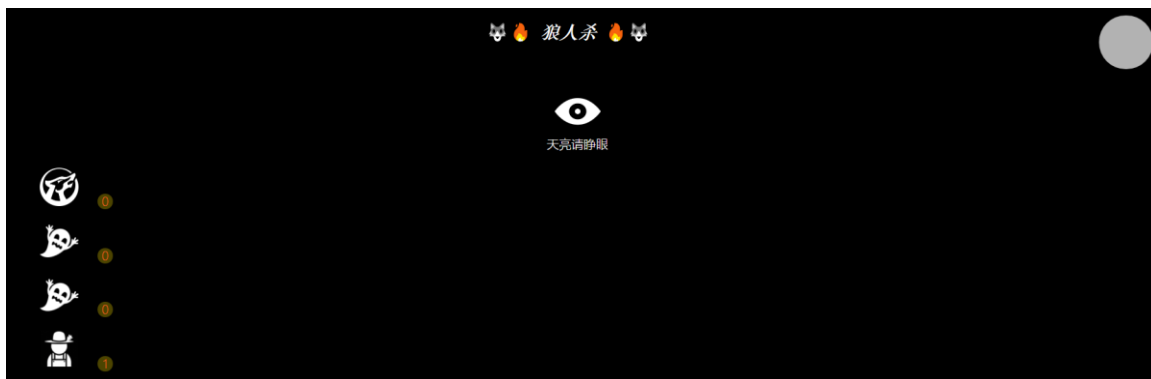


- 投票完成后再次天黑闭眼，循环往复直到游戏结束。

法官宣布天黑请闭眼，白天被投为狼人者，死。



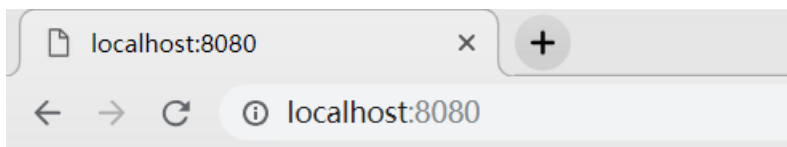
狼人接着投票



法官宣布天亮。（这是另测试的一局游戏，村民胜了）



•此时再次输入 localhost:8080。



本轮游戏已结束，村民胜！

测试

• 开始界面会对地址和数字进行实时检验



法官

地址:

地址不能为空!



法官

地址:

地址不合法!



法官

地址:

人数:

- 如果在非法的情况下强行提交会报错

localhost:8080


localhost:8080 显示

地址不合法!

人数不合法!

请重新填写

确定



法官

地址:

人数:

- 填入人数后下方弹出对应数量输入框

人数:

✓✓✓

add the 1th player

add the 2th player

add the 3th player

add the 4th player

add the 5th player

add the 6th player

开局 清除

- 如果提交了非法的地址会无法正常打开页面



法官

地址:

✓✓✓

人数:

✓✓✓

0xc3d9cc7064e529bbb224ccceab265930a3e:

0xc3d9cc7064e529bbb224ccceab265930a3e:

0xc3d9cc7064e529bbb224ccceab265930a3e:

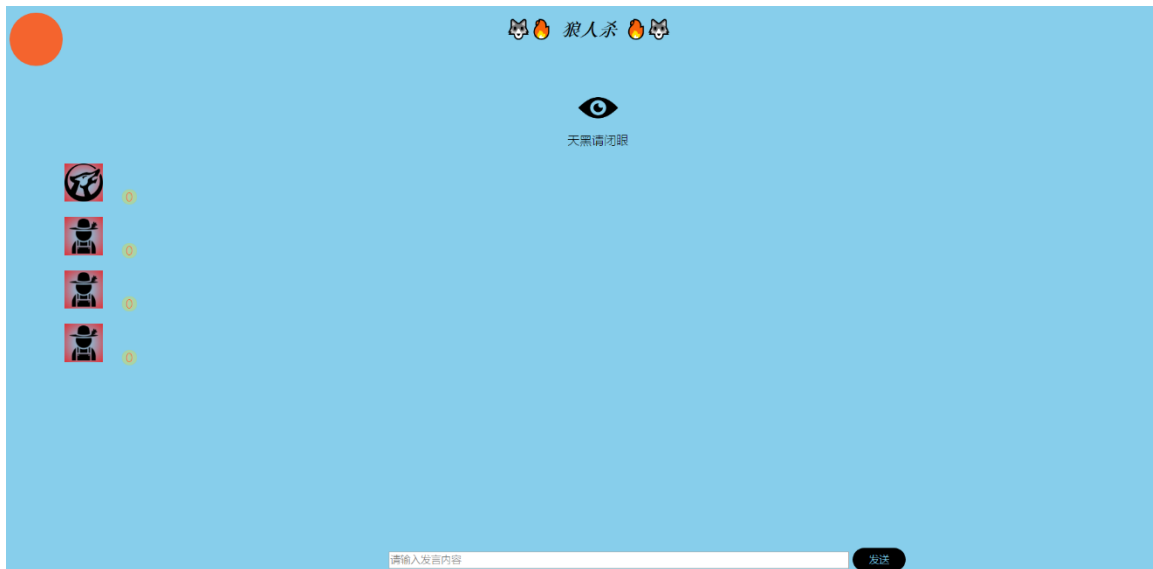
0xc3d9cc7064e529bbb224ccceab265930a3e:

开局 清除

```
Error: sender account not recognized
at Object.InvalidResponse (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\web3\lib\web3\errors.js:38:16)
at RequestManager.send (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\web3\lib\web3\requestmanager.js:61:22)
at Eth.send [as sendTransaction] (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\web3\lib\web3\method.js:145:58)
at SolidityFunction.sendTransaction (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\web3\lib\web3\function.js:170:26)
at C:\Users\13371\killwolf\app.js:62:26
at Layer.handle [as handle_request] (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\express\lib\router\layer.js:95:5)
at next (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\express\lib\router\route.js:137:13)
at Route.dispatch (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\express\lib\router\route.js:112:3)
at Layer.handle [as handle_request] (C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\express\lib\router\layer.js:95:5)
at C:\Program Files (x86)\ethbox\stack\node-v8.11.2-win-x86\node_modules\express\lib\router\index.js:281:22
```

- 在输入正确的前提下可以进入新的页面，红色的头像代表尚未登录。





- 此时法官无法点击天黑请闭眼。



- 法官身份无法为玩家投票。



- 法官无法发言

127.0.0.1:8080 显示

玩家尚未全部登录!

确定

- 游戏开始后玩家登陆输完地址后，会根据不同身份更新头像



狼人

地址: 0xb3cf0b9ac90e416b78f692c0c9f

√√√

加入

清除



村民

地址: 0x8b4a1ee77b52764b3d75af84bt

√√√

加入

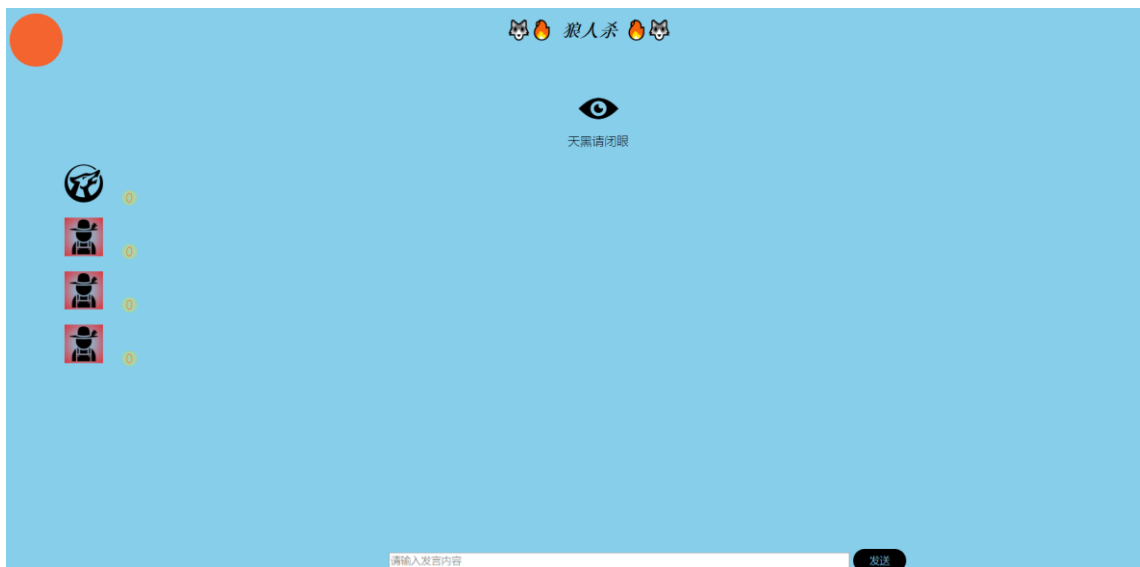
清除



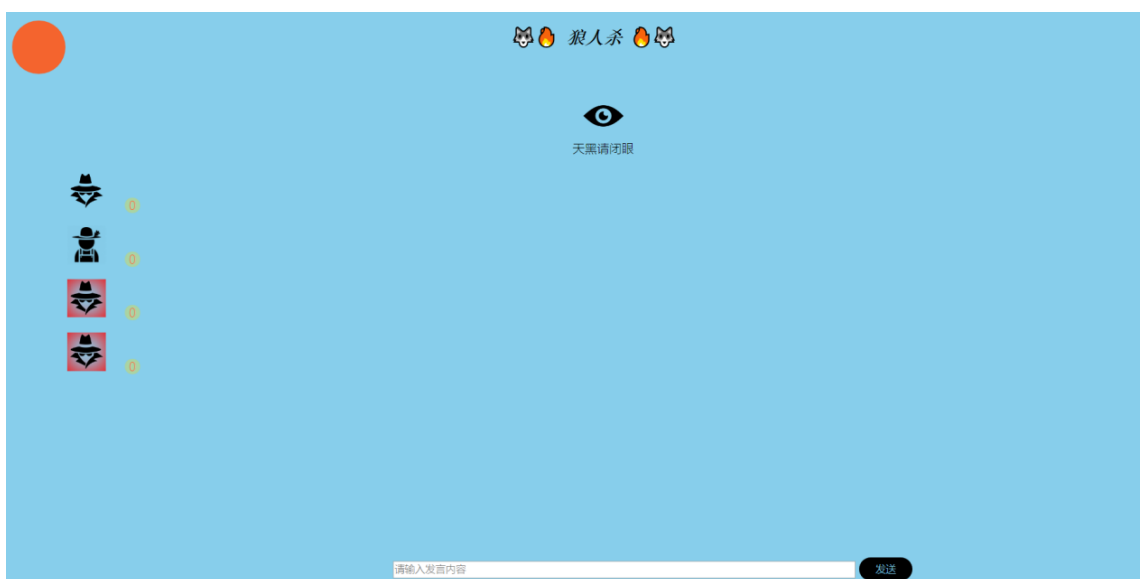
- 输入尚未加入游戏的地址



- 狼人和法官登陆后可以看到所有人的状态。



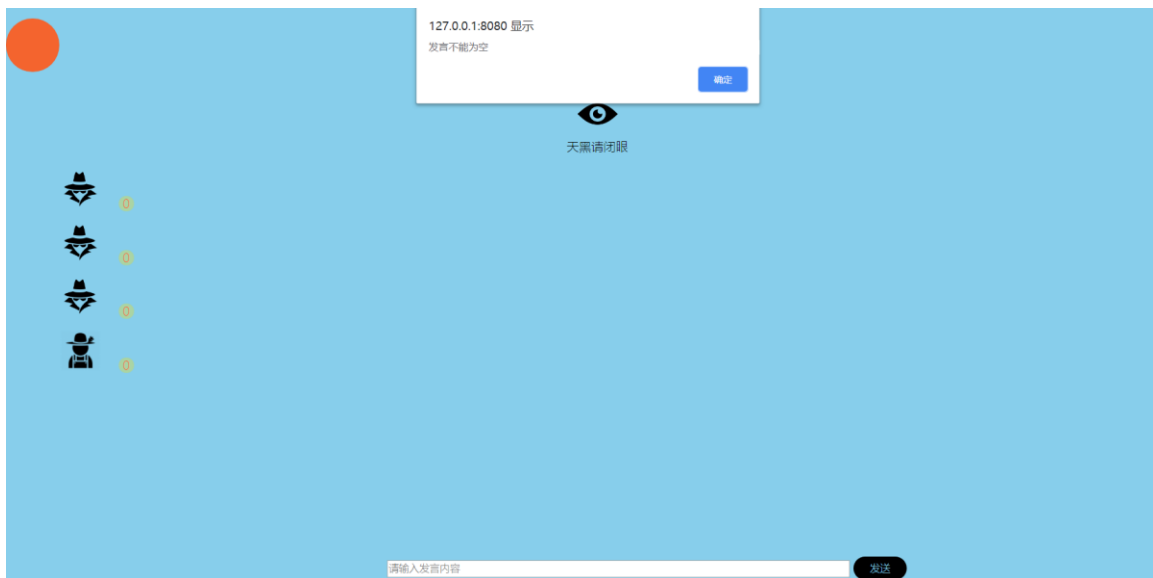
- 村民登陆后只能看到自己的身份和所有人登陆状态。



- 只能由法官启动天黑请闭眼



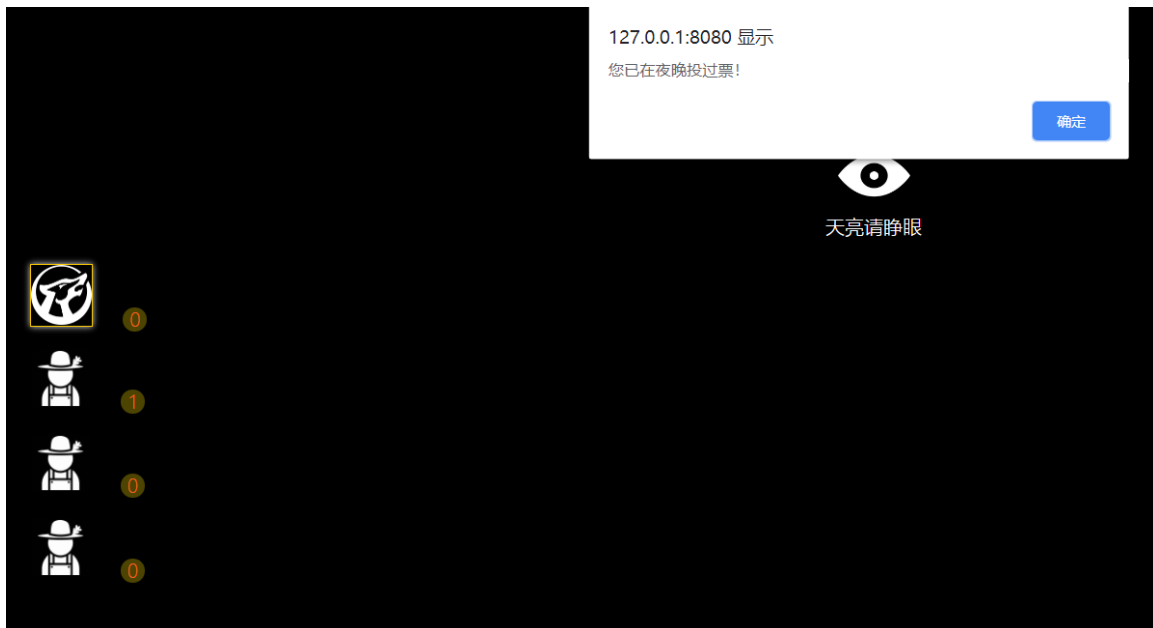
- 发言内容不能为空



- 不能给自己投票



- 狼人只能投一次票



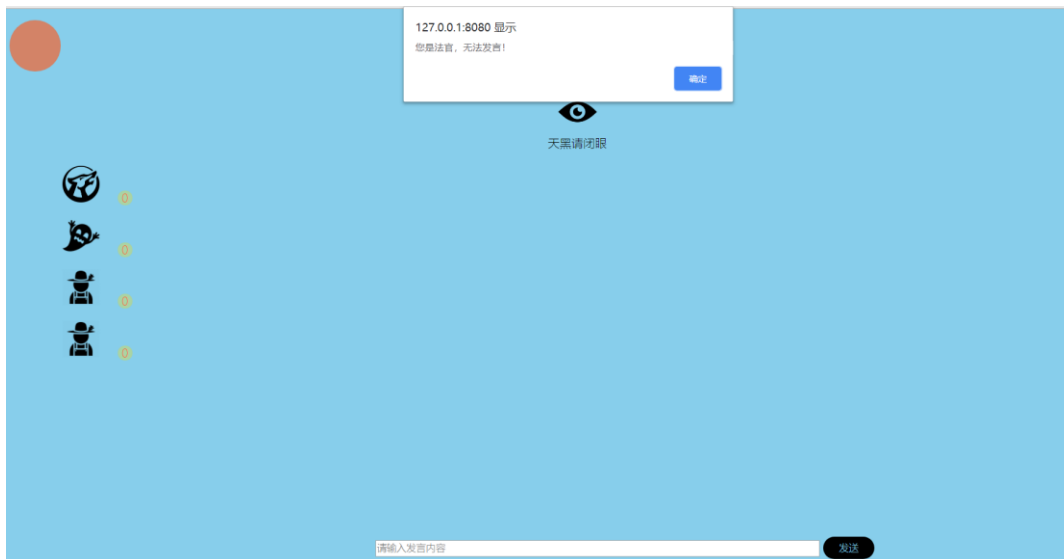
- 狼人无法启动天亮请睁眼



- 村民在夜间无法投票



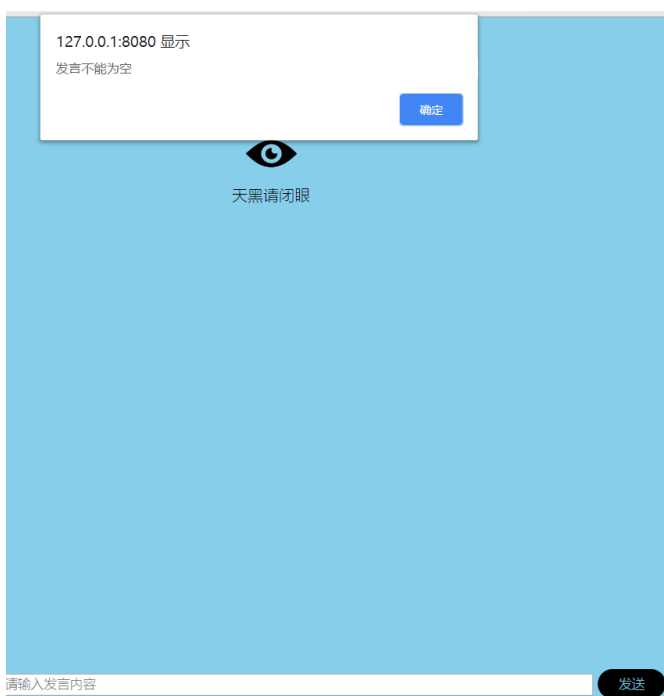
- 白天进入发言环节，法官不得发言



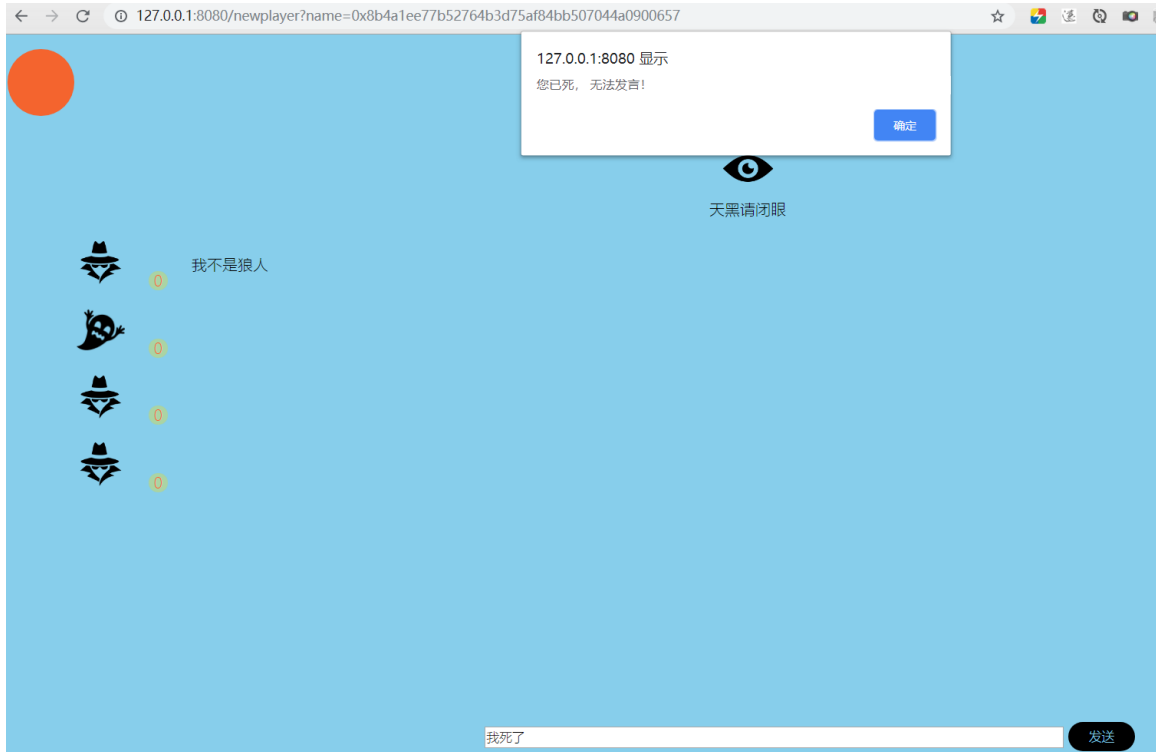
• 法官不得投票



• 发言不得为空



- 死去的人不得发言



- 只能投一次票。



- 死去的人不得给别人投票



- 死去的人的头像是不可点击投票的



- 不能给自己投票



Github 网址（内有动图演示）

<https://github.com/sherluck314/killwolf-using-blockchain>