

RNGG

Random Number Guess Game The blockchain based game on Ethereum with Solidity and Angular

Fabrizio Lazzaretti and Linus Basig

DSA Attestation

University of Applied Sciences Rapperswil (HSR)

University of Applied Sciences of Eastern Switzerland (FHO)

Department of Computer Science

11. December 2019

The RNGG

- ▶ ethereum
- ▶ solidity
- ▶ web interface
- ▶ one parity node per user



Table of Contents

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

Sources

Table of Contents

RNGG Contract Contract

Application

P2P Chat

Testing

Demo

Discussion

The RNGG Process

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract

Contract

Application

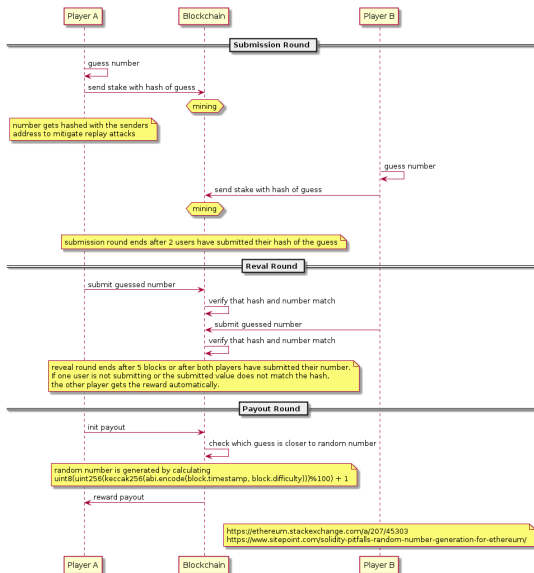
P2P Chat

Testing

Demo

Discussion

Sources



The RNGG Process

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract

Contract

Application

P2P Chat

Testing

Demo

Discussion

Sources

What is the problem?

The RNGG Process

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract

Contract

Application

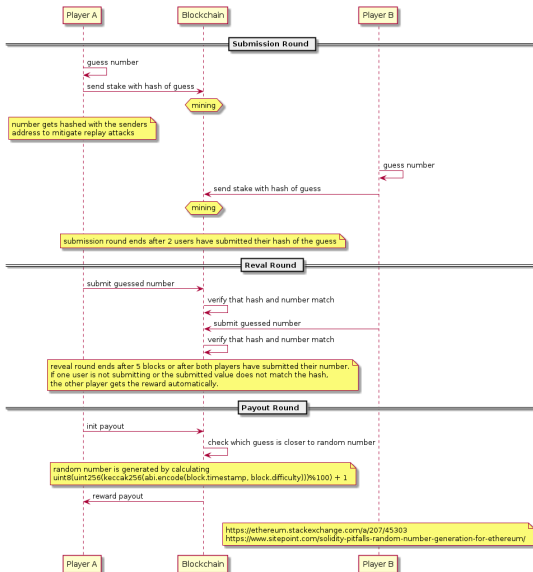
P2P Chat

Testing

Demo

Discussion

Sources



Fabrizio Lazzaretti
and Linus Basig

Contract

```

1 pragma solidity ^0.5.0;
2
3 // https://pastebin.com/pbWBpvCJ
4 contract RandomGuess {
5     struct Game{
6         address player_a;
7         address player_b;
8         uint start;
9         bytes32 hash_a;
10        bytes32 hash_b;
11        uint random;
12        uint num_a;
13        uint num_b;
14    }
15
16    mapping (uint256 => Game) private games;
17    mapping (address => uint256) private user_game;
18    uint256 filling_game = 0;
19    uint256 counter = 0;
20
21    function random() private view returns (uint8) {
22        return uint8(uint256(keccak256(abi.encode(block.timestamp, block.
            difficulty))))%100) + 1;

```


The Discovery Contract II

RNGG

Fabrizio Lazzaretti
and Linus Basig

[Introduction](#)

[RNGG Contract](#)

[Contract](#)

[Application](#)

[P2P Chat](#)

[Testing](#)

[Demo](#)

[Discussion](#)

[Sources](#)

```
23 }
24
25 function abs(int n) private pure returns (int) {
26     if (n >= 0) {
27         return n;
28     }
29     return -n;
30 }
31
32 function min(int a, int b) private pure returns (int) {
33     if (a <= b) {
34         return a;
35     }
36     return b;
37 }
38
39 function getWinner(uint256 game_id) public view returns (address){
40     require(isPayoutReady(game_id));
41     int diff_a_right = abs(int256(games[game_id].random) - int256(
42         games[game_id].num_a));
43     int diff_a_left = abs(int256(games[game_id].random) - int256(
44         games[game_id].num_a) + 100);
45     int diff_a = min(diff_a_right, diff_a_left);
```

Fabrizio Lazzaretti
and Linus Basig

Contract

```

44     int diff_b_right = abs(int256(games[game_id].random) - int256(
        games[game_id].num_b));
45     int diff_b_left = abs(int256(games[game_id].random) - int256(
        games[game_id].num_b) + 100);
46     int diff_b = min(diff_b_right, diff_b_left);
47     if(diff_a == diff_b){
48         // the contract win if both have guessed the same
49         return address(0);
50     }
51     if(diff_a > diff_b) {
52         return games[game_id].player_b;
53     }
54     return games[game_id].player_a;
55 }
56
57 function isWinner(uint256 game_id) public view returns (bool){
58     require(isPayoutReady(game_id));
59     return getWinner(game_id) == msg.sender;
60 }
61
62 event PayoutReady(
63     uint256 game_id
64 );
65

```

The Discovery Contract IV

RNGG

Fabrizio Lazzaretti
and Linus Basig

[Introduction](#)

[RNGG Contract](#)

[Contract](#)

[Application](#)

[P2P Chat](#)

[Testing](#)

[Demo](#)

[Discussion](#)

[Sources](#)

```
66 function isPayoutReady(uint256 game_id) public view returns (bool){
67     // todo should also be possible after n blocks
68     // todo should only be executed once per game
69
70     if(games[game_id].num_a > 0 && games[game_id].num_b > 0){
71         return true;
72     }
73     return false;
74 }
75
76 event RevalReady(
77     uint256 game_id
78 );
79
80 function isRevalReady(uint256 game_id) public view returns (bool){
81     if(games[game_id].hash_a != "" && games[game_id].hash_b != ""){
82         return true;
83     }
84     return false;
85 }
86
87 function computeSecret(uint8 number) public view returns (bytes32) {
88     require(number > 0 && number <= 100);
```

Fabrizio Lazzaretti
and Linus Basig

```

89         return keccak256(abi.encode(number, msg.sender));
90     }
91
92     constructor() public {
93
94     }
95
96     function nextCounter() private returns (uint256) {
97         return ++counter;
98     }
99
100    function register(bytes32 hash) payable external returns (uint256) {
101        require(msg.value == 2 wei);
102
103        uint256 game_id = 0;
104        if (filling_game == 0) {
105            uint256 id = nextCounter();
106            games[id] = Game(msg.sender, address(0), now, hash, "", 0,
107                           0, 0);
108            filling_game = id;
109            game_id = id;
110        } else {
111            games[filling_game].player_b = msg.sender;

```

Contract

Fabrizio Lazzaretti
and Linus Basig

Contract

```

111         games[filling_game].hash_b = hash;
112         games[filling_game].random = random();
113         game_id = filling_game;
114         filling_game = 0;
115     }
116
117     if (isRevalReady(game_id)) {
118         emit RevalReady(game_id);
119     }
120
121     user_game[msg.sender] = game_id;
122
123     return game_id;
124 }
125
126 function getMyGame() public view returns (uint256) {
127     return user_game[msg.sender];
128 }
129
130 function submitReval(uint256 game_id, uint num) public returns (bool) {
131     {
132         require(isRevalReady(game_id) && num > 0 && num <= 100);
133         // todo check if num is possible, according to submitted hash

```

The Discovery Contract VII

RNGG

Fabrizio Lazzaretti
and Linus Basig

Contract

```

133         if (games[game_id].player_a == msg.sender){
134             games[game_id].num_a = num;
135         } else if (games[game_id].player_b == msg.sender){
136             games[game_id].num_b = num;
137         } else{
138             return false;
139         }
140     }
141
142     // todo why does this not work???
143     // if(games[game_id].num_a > 0 && games[game_id].num_b > 0){
144         //     emit PayoutReady(game_id);
145     // }
146
147     return true;
148 }
149
150 function sendMoney(uint256 game_id) public payable {
151     // todo should only be executed once per game
152     require(isPayoutReady(game_id));
153     address winner = getWinner(game_id);
154
155     if (winner != address(0)){

```

Fabrizio Lazzaretti
and Linus Basig

Contract

```

156     address payable addr = address(uint160(winner));
157     if (address(addr).send(4 ether)) {
158         // done
159     }
160 }
161 }
162
163 function getGameInfo(uint256 game_id) public view returns (address,
164     uint256, address, uint256, uint256) {
165     return (games[game_id].player_a,
166         games[game_id].num_a,
167         games[game_id].player_b,
168         games[game_id].num_b,
169         games[game_id].random);
170 }
171 }

```

RNGG Contract

Application

P2P Chat

Testing

Demo

Discussion

Application

- ▶ Angular 8
- ▶ RxJS for State Management



```
src
├── app
│   ├── app.component.html
│   ├── app.component.scss
│   ├── app.component.spec.ts
│   ├── app.component.ts
│   ├── app.module.ts
│   ├── app-routing.module.ts
│   └── chat
│       ├── chat.component.html
│       ├── chat.component.scss
│       ├── chat.component.spec.ts
│       └── chat.component.ts
├── eth.service.ts
├── info-messages.service.spec.ts
├── info-messages.service.ts
├── peer.service.spec.ts
├── peer.service.ts
├── random-game
│   ├── random-game.component.html
│   ├── random-game.component.scss
│   ├── random-game.component.spec.ts
│   ├── random-game.component.ts
│   ├── random-game.service.spec.ts
│   └── random-game.service.ts
├── settings
│   ├── settings.component.html
│   ├── settings.component.scss
│   ├── settings.component.spec.ts
│   └── settings.component.ts
├── test-contract
│   ├── test-contract.component.html
│   ├── test-contract.component.scss
│   ├── test-contract.component.spec.ts
│   ├── test-contract.component.ts
│   ├── test-contract.service.spec.ts
│   └── test-contract.service.ts
├── assets
│   └── contract
│       └── kovan.json
├── environments
│   ├── environment.prod.ts
│   └── environment.ts
├── favicon.ico
├── index.html
├── main.ts
├── polyfills.ts
├── styles.scss
└── test.ts
```

Table of Contents

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

Sources

RNGG

P2P Chat

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻ 20/30

```
1 pragma solidity ^0.5.0;
2
3 contract Signaling {
4     event NewPartner(
5         bytes16 id
6     );
7
8     function triggerMe(bytes16 id) public {
9         emit NewPartner(id);
10    }
11 }
```

Table of Contents

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

Sources

Test Setup

- ▶ docker-compose
- ▶ deployment script

CI Testing

RNGG

Fabrizio Lazzaretti
and Linus Basig

[Introduction](#)

[RNGG Contract](#)

[Contract](#)

[Application](#)

[P2P Chat](#)

[Testing](#)

[Demo](#)

[Discussion](#)

[Sources](#)

► GitLab CI

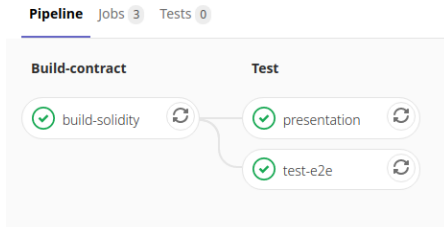


Table of Contents

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract
Contract

Application

P2P Chat

Testing

Demo

Discussion

Sources

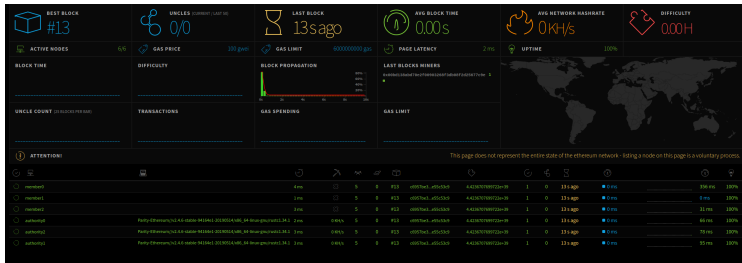
Demo setup

RNGG

Fabrizio Lazzaretti
and Linus Basig

Contract

Demo



Demo: RNGG

RNGG

Fabrizio Lazzaretti
and Linus Basig

Introduction

RNGG Contract

Contract

Application

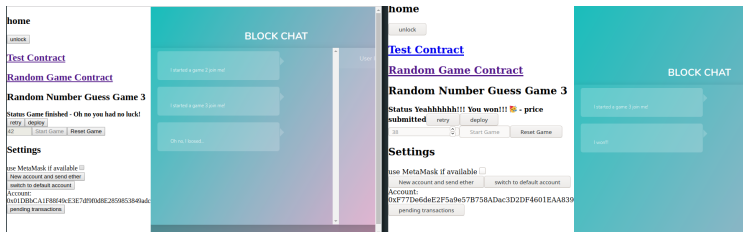
P2P Chat

Testing

Demo

Discussion

Sources



RNGG Contract

Application

P2P Chat

Testing

Demo

Discussion

Discussion

- ▶ good test setup
- ▶ web3.js and parity is not stable
- ▶ webRTC signaling server is also not perfect
- ▶ rust for WASM was not possible

Sources

- ▶ Repository:
`https://gitlab.com/block-game/
angular-web3-smartcontract`