Blockchain and Cryptocurrencies Project Final Milestone

Divij Singh

06/05/19

1 Introduction

The idea behind this project is to create a betting platform for the video game League of Legends.
The game frequently hosts official professional tournaments, with several teams from the world over competing. These matches have followings rivaling that of more traditional sports, and as such, betting would be of interest.

Currently, a few platforms for betting exist, but they all consist of a centrally hosted server. This is insecure, due to the risk of back-end modification of bets, potential downtimes and data-loss, or rabid sheep infections.

By hosting this on the blockchain, the risks are all but eliminated, though a sheep apocalypse stays a viable danger.

Additionally, it allows people to avoid those pesky anti-betting laws.

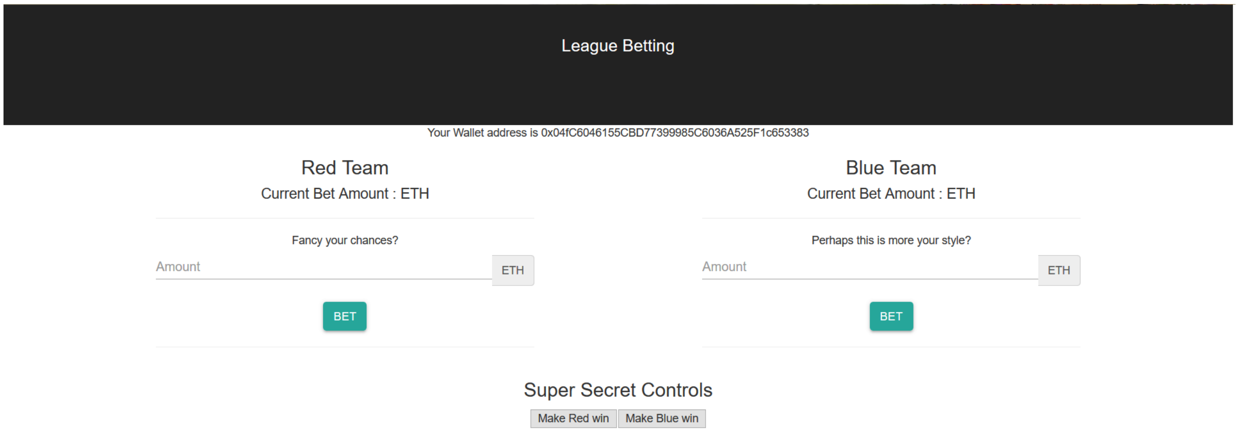
2 Project Description

The project is a simple betting dapp, built on the Etherium's Ropsten test network, using ReactJS as a framework and Truffle to manage contract deployment.

3 Design Specification

The UI and front-end have been created using ReactJS, and Materialize CSS templates.

As the project is still rough, it currently has a single page which allows a person to bet, and has admin controls at the bottom to allow for selection of a winner. Unfortunately, Riot Games API integration was not possible.



Your Wallet address is 0x04fC6046155CBD77399985C6036A525F1c653383

Current Bet Amount : ETH

Fancy your chances?

Current Bet Amount : ETH

Perhaps this is more your style?

Figure 1: What the main page looks like

It is deployed on the Ropsten test network, using an Infura endpoint. Account management is handled via the Metamask extension for Firefox/Google Chrome.

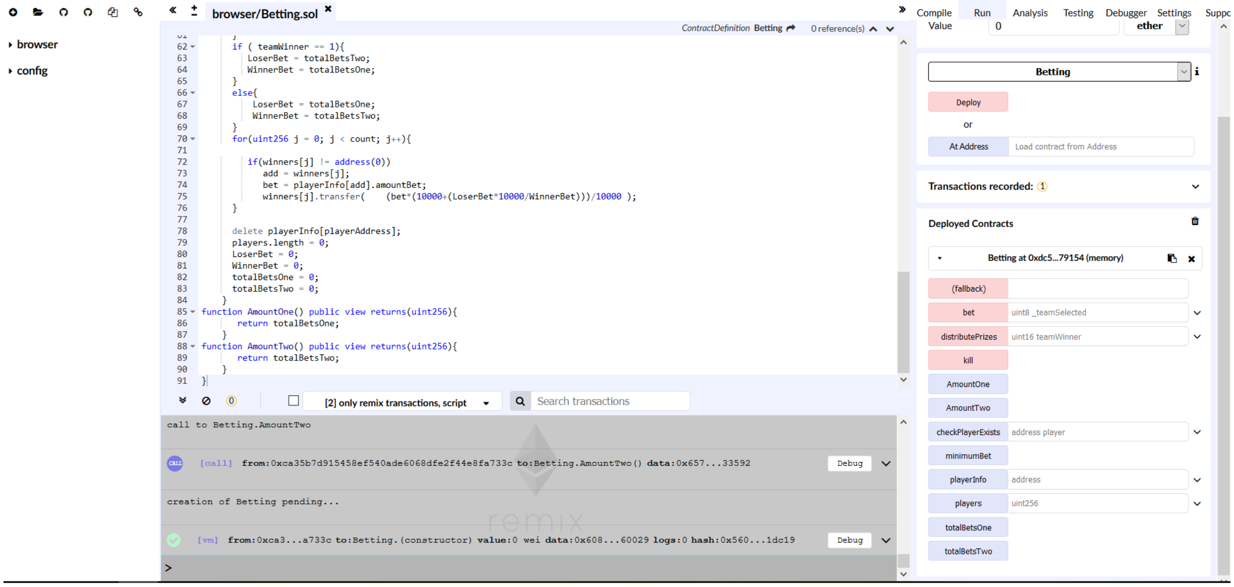
Truffle and Web3 ar utilized to both deploy and interact with deployed smart contracts.

4 Workflow

First, the admin deploys the contract via the command-line.
Then, people can bet on either team using the front-end page. Transactions are handled via the Metamask Etherium Waller browser extension.
Once a victor is decided, the admin can click on the corresponding button for the winning team, and send the rewards out.

5 Implementation

In terms of what has been implemented, technically, everything should work.
But, as is life in computer science, things do not. The smart contract is fully functional; it has been tested on the online Remix IDE, and has been successfully deployed to the Ropsten network via Truffle itself, as can be seen in the



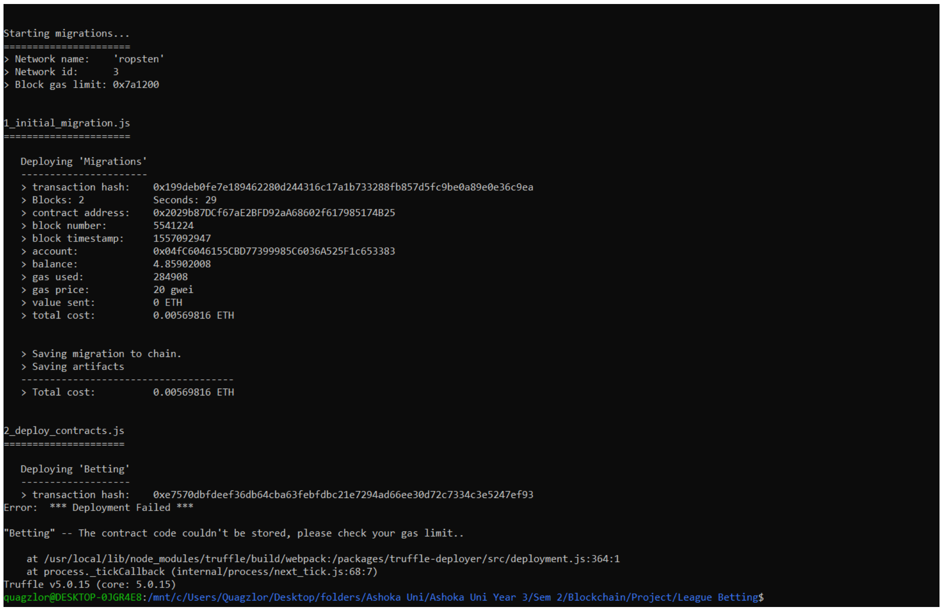
```
        if ( teamWinner == 1){
           LoserBet = totalBetsTwo;
           WinnerBet = totalBetsOne;
        }
        else{
            LoserBet = totalBetsOne;
            WinnerBet = totalBetsTwo;
        }
        for(uint256 j = 0; j < count; j++){

           if(winners[j] != address(0))
              add = winners[j];
              bet = playerInfo[add].amountBet;
              winners[j].transfer(    (bet*(10000+(LoserBet*10000/WinnerBet)))/10000 );
        }

        delete playerInfo[playerAddress];
        players.length = 0;
        LoserBet = 0;
        WinnerBet = 0;
        totalBetsOne = 0;
        totalBetsTwo = 0;
      }
function AmountOne() public view returns(uint256){
         return totalBetsOne;
      }
function AmountTwo() public view returns(uint256){
         return totalBetsTwo;
      }
}
```

```
call to Betting.AmountTwo

[call] from:0xca35b7d915458ef540ade6068dfe2f44e8fa733c to:Betting.AmountTwo() data:0x657...33592

creation of Betting pending...

[vm] from:0xca3...a733c to:Betting.(constructor) value:0 wei data:0x608...60029 logs:0 hash:0x560...1dc19

>
```

Figure 2: The functions of the contract can be seen on the right

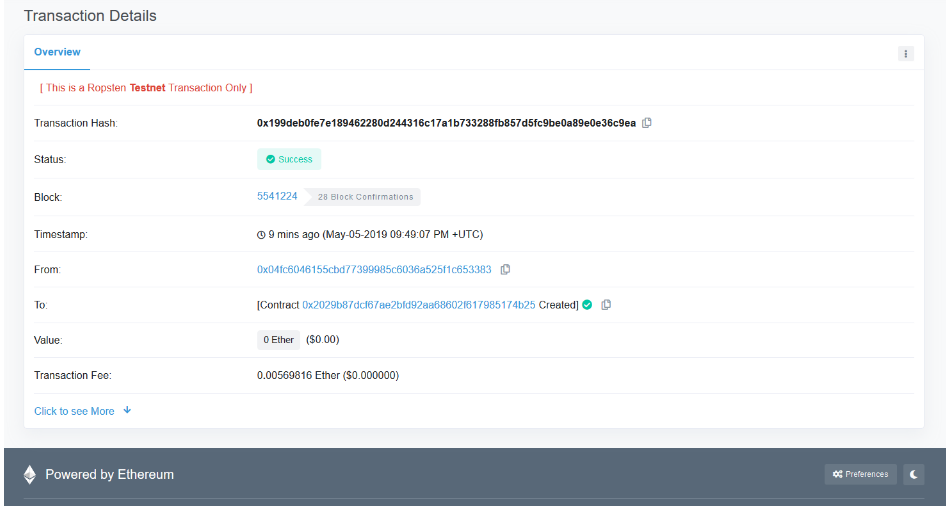
following images:

Unfortunately, I have been unable to retrieve the contract data via the front-end; as such, the website does not interact (hence the fact that admin controls are right on the front page)



```
Starting migrations...
======================
> Network name:    'ropsten'
> Network id:      3
> Block gas limit: 0x7a1200


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0x199deb0fe7e189462280d244316c17a1b733288fb857d5fc9be0a89e0e36c9ea
   > Blocks: 2            Seconds: 29
   > contract address:    0x2029b87DCf67aE2BFD92aA68602f617985174B25
   > block number:        5541224
   > block timestamp:     1557092947
   > account:             0x04fC6046155CBD77399985C6036A525F1c653383
   > balance:             4.85902008
   > gas used:            284908
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00569816 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00569816 ETH


2_deploy_contracts.js
=====================

   Deploying 'Betting'
   -------------------
   > transaction hash:    0xe7570dbfdeef36db64cba63febfdbc21e7294ad66ee30d72c7334c3e5247ef93
Error:  *** Deployment Failed ***

"Betting" -- The contract code couldn't be stored, please check your gas limit..

    at /usr/local/lib/node_modules/truffle/build/webpack:/packages/truffle-deployer/src/deployment.js:364:1
    at process._tickCallback (internal/process/next_tick.js:68:7)
Truffle v5.0.15 (core: 5.0.15)
quagzlor@DESKTOP-0JGR4E8:/mnt/c/Users/Quagzlor/Desktop/folders/Ashoka Uni/Ashoka Uni Year 3/Sem 2/Blockchain/Project/League Betting$
```

Figure 3: Successful migration and deployment of a contract to the Ropsten network



Transaction Details

[This is a Ropsten **Testnet** Transaction Only]

Transaction Hash:	**0x199deb0fe7e189462280d244316c17a1b733288fb857d5fc9be0a89e0e36c9ea**
Status:	✓ Success
Block:	5541224 28 Block Confirmations
Timestamp:	9 mins ago (May-05-2019 09:49:07 PM +UTC)
From:	0x04fc6046155cbd77399985c6036a525f1c653383
To:	[Contract 0x2029b87dcf67ae2bfd92aa68602f617985174b25 Created] ✓
Value:	0 Ether ($0.00)
Transaction Fee:	0.00569816 Ether ($0.000000)

Figure 4: Verified via Etherscan