

Challenge Task 2018

Implementation of a Decentralized Application Tic Tac Toe

Departements of Informatics - Communication Systems Group, Chair

Lucas Pelloni, leginumeber
Severin Wullschleger leginumber
Andreas Schaufelbühl, 12-918-843



**University of
Zurich**^{UZH}



TABLE OF CONTENTS

	Page
LIST OF TABLES	iv
LIST OF FIGURES	v
CHAPTER	
1 Introduction	1
2 Technologies	2
2.1 Solidity	2
2.2 Web3.js	2
2.3 MetaMask	2
2.4 Ganache	2
3 Implementation of the game	3
3.1 Project Structure	3
3.2 Game Walk-through	4
4 Discusion	5
4.1 Challenges and Problems	5
4.2 Future work	5
A Raw Data	6
REFERENCES	6
APPENDIX	

LIST OF TABLES

Table	Page
-------	------

LIST OF FIGURES

Figure	Page
3.1 Project Structure with Technologies	3
3.2 Class-Diagram of the Smart Contract	4

Chapter 1

INTRODUCTION

This years Challenge Task is to implement a Decentralized Application (DApp) running in the Ethereum blockchain. The goal of the application is a playable Tic-Tac-Toe¹ game, which also includes a betting system, all embedded in a Smart Contract.

Chapter 2 gives an overview and short explanation of the technologies we use in order to implement the Challenge Task.

In Chapter 3 we show the actual implementation of the game. It starts by explaining and showing our project structure. Also we give walk-through of the different processes of playing a game and betting on games.

The problems and challenges occurred within our project are discussed in Chapter 4. Additionally we also describe our open task and goals for the future concerning this project.

¹<https://en.wikipedia.org/wiki/Tic-tac-toe>

Chapter 2

TECHNOLOGIES

With Solidity ¹ we implement the smart contract which will run on the Ethereum blockchain platform. For our front-end we choose using React ², which is a JavaScript library for building user interfaces. The interaction of the front-end application with our smart-contract is provided through Web3.js ³ and MetaMask ⁴. To speed up the testing and development we use Ganache ⁵ to run our local Ethereum blockchain. In the following section we describe the different technologies and its use in our project more in detail.

2.1 Solidity

2.2 Web3.js

2.3 MetaMask

2.4 Ganache

¹<https://github.com/ethereum/solidity>

²<https://reactjs.org/>

³<https://web3js.readthedocs.io/en/1.0/>

⁴<https://metamask.io/>

⁵<http://truffleframework.com/ganache/>

Chapter 3

IMPLEMENTATION OF THE GAME

3.1 Project Structure

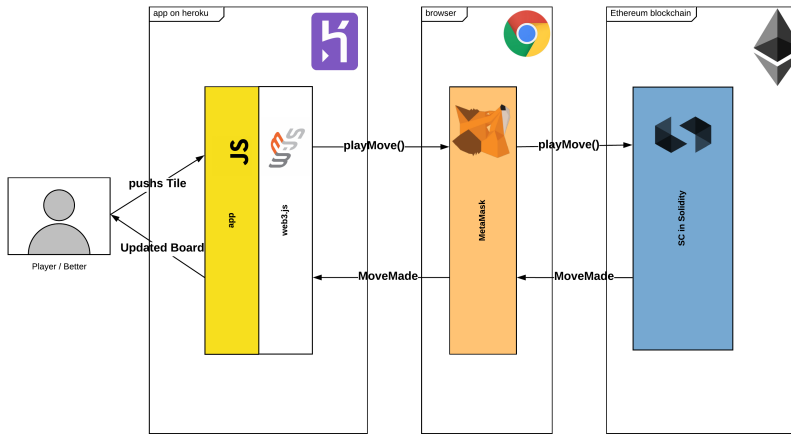


Figure 3.1: Project Structure with Technologies

Figure 3.1 shows the project structure and the interaction between the different systems by the example of an Player choosing a tile on the board. The web-application is running on the Heroku Platform ¹. Through the browser and MetaMask a User can get verified by its Ethereum-account and pay the requested amount of gas in order to run functionalities on the Ethereum smart contract. The smart contract itself runs on an blockchain, which can be either a private or the Ropsten Testnet ².

¹<https://www.heroku.com/>

²<https://ropsten.etherscan.io/>

The smart contract firstly checks if the move is valid. Secondly it looks for a winner and changes the game state if so. After that it returns a move confirmation to the user.

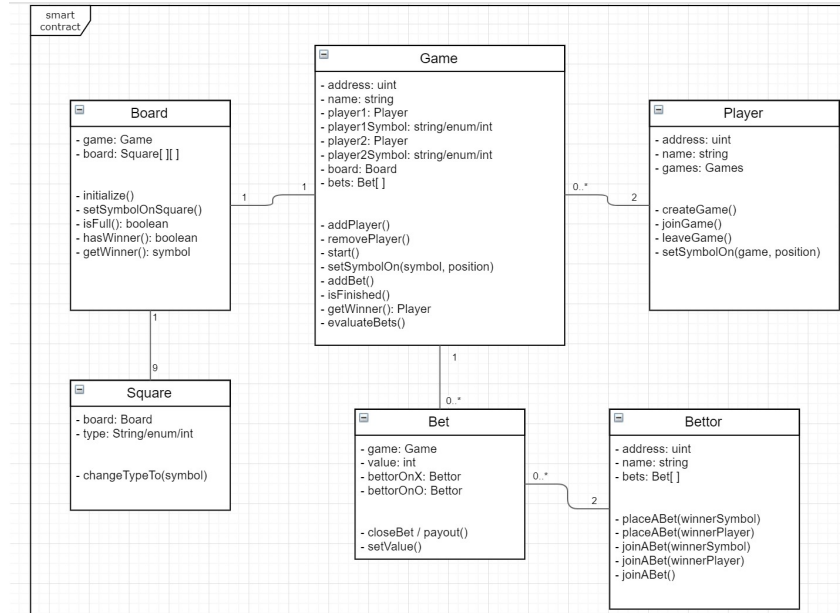


Figure 3.2: Class-Diagram of the Smart Contract

3.2 Game Walk-through

Chapter 4

DISCUSSION

4.1 Challenges and Problems

4.2 Future work

APPENDIX A
RAW DATA