DApp for the NFT TRY Lottery

Finalterm for Peer to Peer & Blockchain 21-22

Chenxiang Zhang
Department of Computer Science
University of Pisa
`c.zhang4@studenti.unipi.it`

1 Introduction

In this didactic blockchain project, we continue to extend the second midterm by creating a DApp for the TRY Animal Lottery. We invite the readers to refresh the architecture of the Lottery backend with the report presented in the second midterm. The purpose of this project is to build a DApp that interacts with the lottery backend by implementing two different interfaces, one for the administrator of the lottery and one for all the players. The DApp must also catch the events emitted by the smart contract backend.

2 DApp Lottery Architecture

From Figure 1, we can see that the DApp has three different states: Start part and Lottery part (admin or player). The former is the landing page when launching the DApp, from this state the only available action is to click on the "CREATE LOTTERY" button which will change the page status to the Lottery part. When in the Lottery part, it is possible to interact with the core of the Lottery game (Admin Actions or Buy Ticket) and change from admin and player mode. We describe in detail the two parts in the following subsections.

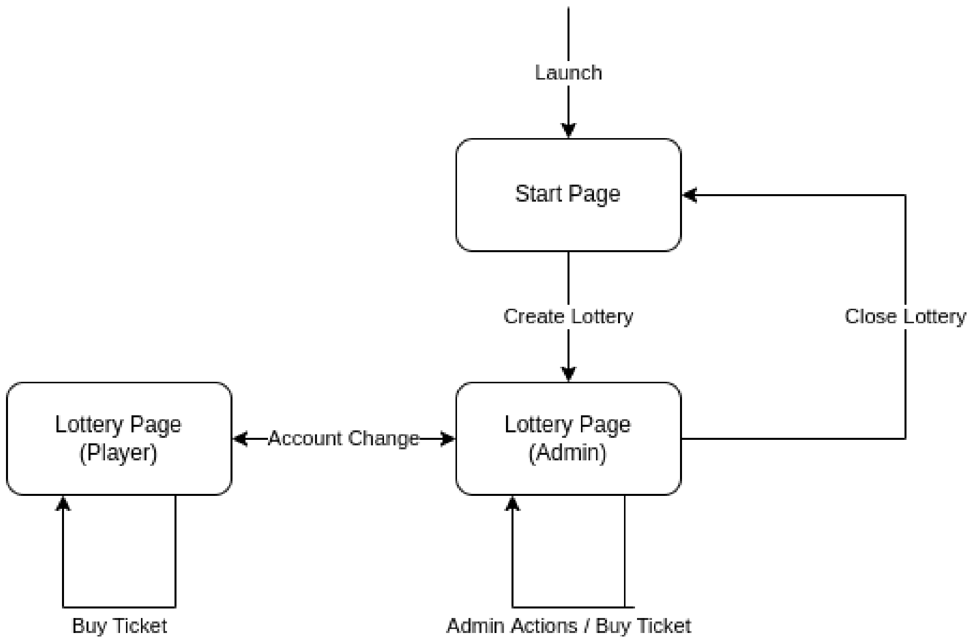


Figure 1: Architecture of the DApp

2.1 Start Part

This part represents the starting point when launching the DApp. Figure 2 shows the actual page of the application. It contains some of the basic information such as the user's public address and its available balance in Ethereum. The single available action is the button "CREATE LOTTERY", which will transition the system to the Lottery Part.

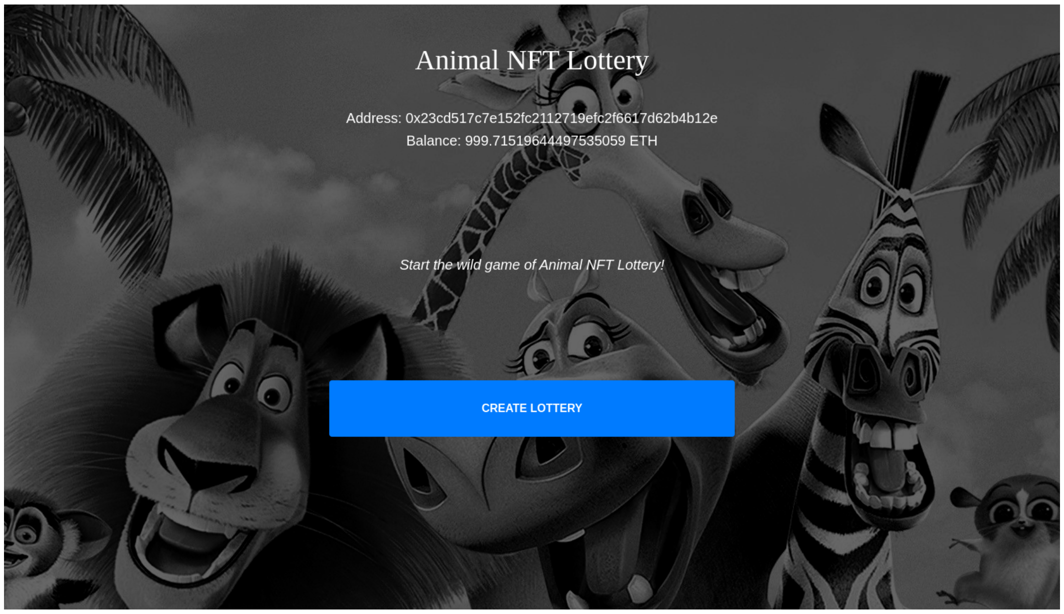
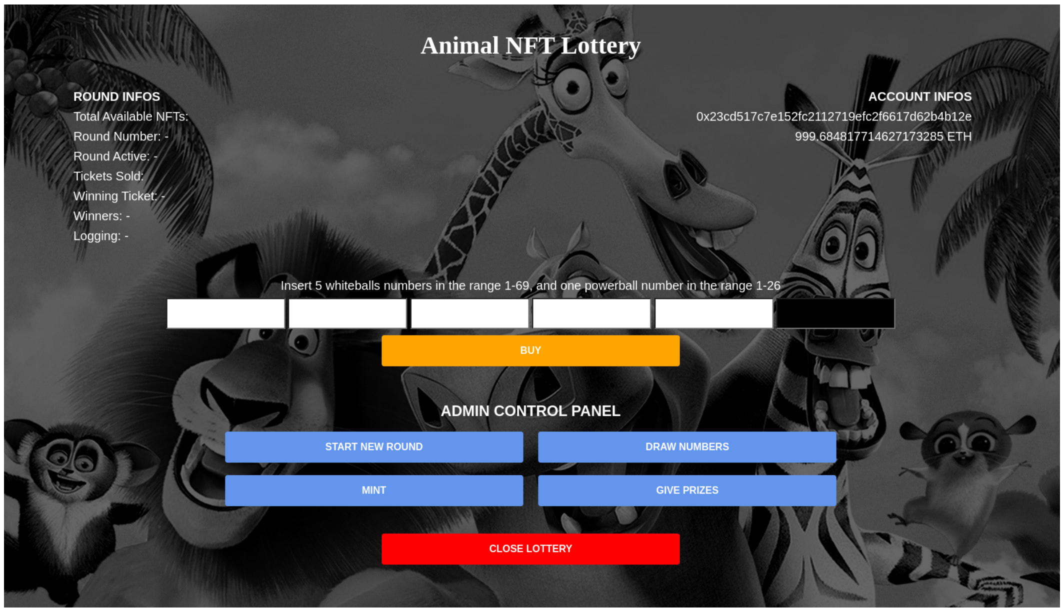

Animal NFT Lottery
Address: 0x23cd517c7e152fc2112719efc2f6617d62b4b12e
Balance: 999.71519644497535059 ETH
Start the wild game of Animal NFT Lottery!
CREATE LOTTERY

Figure 2: Start part: landing page when launching the DApp

2.2 Lottery Part

This part of the DApp represents the core of the game. Figure 3 shows the actual interface of the lottery part. It contains four main sections:

1. *Account Informations.* As in the Starting page, this section reports the current user's address and its balance in ETH. This section is updated every time the balance is changed or when there's a account chnage in Metamask.
2. *Round Informations.* This section contains all the informations related to the current round's state. It shows the total number of NFT available as prizes, the number of the current round, the status of the round, the list of sold tickets of the current round, the winning extracted ticket, and the list of the potential winners of the round associated with the NFT prize assigned.
3. *Ticket Purchcase.* This section allows any user to buy a ticket for the lottery. There are 6 input forms allowing the input of 5 whiteball and 1 powerball numbers to partecipate in the round by clicking the BUY button and paying the ticket price. The purchase will be successful only if the numbers are in the range (1-69 for whiteballs, 1-26 for powerball), if the whiteballs don't contain any duplicates, and the round status must be active.
4. *Admin Control Panel.* This section is only visible when the current user is the admin of the lottery. It contains the main button functions to interact with the lottery backend manipulating the state of the lottery. The functions are: START NEW ROUND, DRAW NUMBERS, MINT, GIVE PRIZES, and CLOSE LOTTERY.


Animal NFT Lottery

ROUND INFOS
Total Available NFTs:
Round Number: -
Round Active: -
Tickets Sold:
Winning Ticket: -
Winners: -
Logging: -

ACCOUNT INFOS
0x23cd517c7e152fc2112719efc2f6617d62b4b12e
999.684817714627173285 ETH

Insert 5 whiteballs numbers in the range 1-69, and one powerball number in the range 1-26

Figure 3: Lottery part: the core of the DApp

The DApp must also catch all the emitted events from the Lottery smart contract. The events are catched by the DApp using the "listenForEvents" function. For any event emitted by the lottery backend, we have a visual feedback that changes a section of our DApp and we log in the console the results. The available event listeners and their relative effects are:

1. *Logging.* General logging event that contains a string message, it can contain anything. The listener will display the information in *Round Informations - Logging*.
2. *LoggingTicketPurchase.* For every ticket purchase, we will update the section *Round Informations - Tickets Sold* to display and append the information regarding the address of the player who purchased the ticket and its ticket numbers.
3. *LoggingTicketWin.* After the winning ticket is extracted, this event is emitted and its listener will display the winning numbers in the section *Round Informations - Winning Ticket*.
4. *LoggingMintNFT.* For every new minted NFT, this event will be emitted and its listener will display the information about the new NFT in *Round Informations - Logging*.
5. *LoggingWinner.* For every player that satisfy the winning condition, at the end of the round, we update the *Round Informations - Winners* to append the winner and its NFT prize to the list of winners.
6. *accountsChanged.* This event is emitted for every change of Metamask account, we update the UI of the DApp to enable/disable the *Admin Control Panel* section depending on the current account's address.

3 Implementation Details

The interaction with the Lottery backed is managed with the use of asynchronous calls. Web3 calls to smart contract functions are asynchronous, and implemented by Promise objects in Javascript. For simplicity we use the sync and await syntax that results to a complete synchronous code. The deployment of the contracts is on the local network by using Ganache [1].

[1]https://trufflesuite.com/ganache/

4 Conclusion

In this didactic blockchain project, we build a lottery smart contract from scratch using the Solidity programming language on the Ethereaum blockchain. Then we build an DApp to interact with the smart contract deploying everything on the local network using Ganache. We have learned about many cutting-edge technologies, using different tools, and applying knowledge gained from the classes. It was a valuable project to start approaching this vast, interesting, and economically crazy ecosystem of blockchain.