

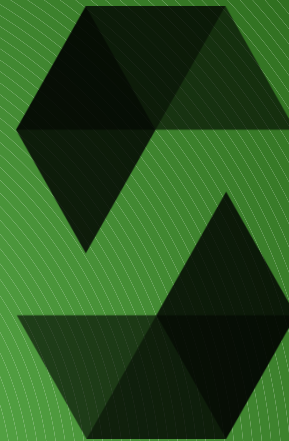
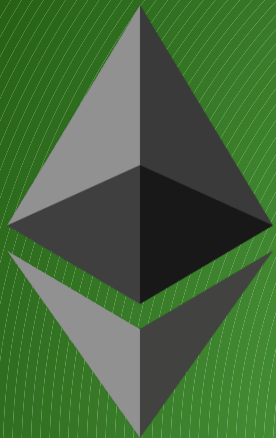
# Фер игра на срећу имплементирана коришћењем Ethereum паметних уговора

Студент:  
Ђорђе Гачић 626/2018

Професор:  
др. Владимир Миловановић

Крагујевац, септембар 2021.

# Средства за имплементацију





# Логика паметних уговора

- Blockchain
- Solidity програмски језик (верзија 0.8.7)
- Сопствена адреса
- Уграђивање у blockchain
- Немогуће измјенити код
- Увијек се извршава исти код

# Функционисање лутрије

- Креирање сопствене лутрије
- Придруживање већ креираној лутрији
- Свака креирана лутрија посједује идентификатор
- Критеријум за учешће (улог мора бити већи или једнак од просјечног улога)
- На адреси уговора је и листа лутрија
- Лутрија садржи листу опклада
- Креатор је власник прве опкладе у креираној лутрији



# Креирање лутрије

- Унос адресе корисника (Ethereum адреса) – 160 бита
- Унос броја учесника
- Унос вриједности улога (Ether валута)
- Унос приватног кључа – 256 бита
- Потписивање трансакције и креирање лутрије

# Придруживање лутрији

- Унос адресе корисника (Ethereum адреса) – 160 бита
- Унос идентификатора лутрије (lottery ID)
- Унос вриједности улога (Ether валута)
- Унос приватног кључа – 256 бита
- Потписивање трансакције и додавање опкладе унутар лутрије



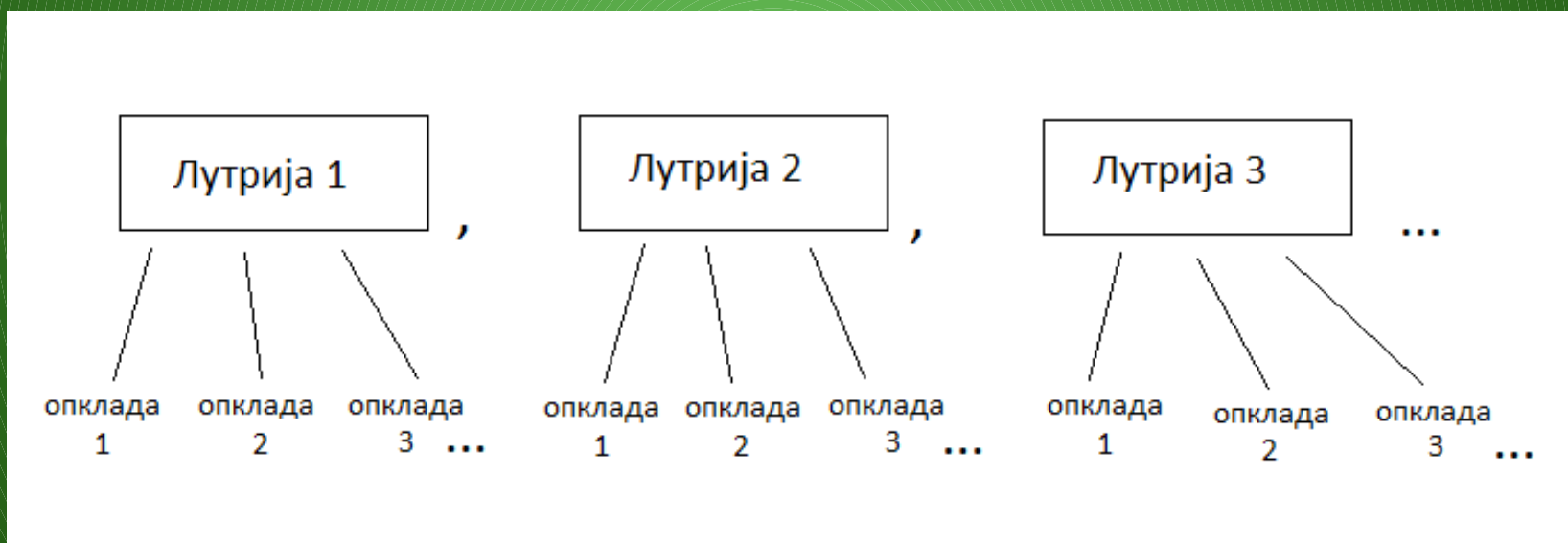
# Извлачење добитника

- Услов: придружено тачно онолико опклада колико је предвидио креатор лутрије
- Не постоји погађање било каквих вриједности!
- Исход лутрије је добитни индекс
- Свака опклада има сопствени индекс у низу
- Хеш функција кесак256
- Улаз: низ опклада, адреса рудара, timestamp блока
- Излаз: 256 битна вриједност
- Конвертовање у цијели број
- Остатак при дијелењу са бројем опклада = добитни индекс

# Опис паметног уговора

Адреса уговора:

0xb6eF88560d255bA8766462F161f415160613FC  
02





# Опис паметног уговора

- Класа Gambling
- Структура Bet
- Структура Game
- Функција createGame
- Функција takeBet
- Функција payout
- Функција generateGameOutcome
- Функција checkPermissions

# Опис паметног уговора

- Функција `getLastID`
- Функција `getGameValue`
- Функција `getCurrentNumOfBets`
- Функција `getAmountToEnterGame`
- Функција `getGameCapacity`
- Функција `getGameStatus`
- Функција `getCreatorAddress`
- Функција `getCurrentGameBets`
- Функција `getMyIndex`
- Функција `getGameOutcome`



# Генерисање исхода - функција

```
// function to randomly generate game outcome
function generateGameOutcome( uint _gameID) private {
    checkPermissions( msg.sender , _gameID);
    games[_gameID].status = STATUS_COMPLETE;
    Bet[] memory tempArrBets = games[_gameID].bets;
    // generate random number: (array of bets, current block miner's address and current block timestamp as input for hash function)
    games[_gameID].outcome = uint(keccak256(abi.encode(tempArrBets, block.coinbase, block.timestamp)))%games[_gameID].numOfBets;
    // winner is address in bet with index equal to outcome
    for(uint j = 0; j < games[_gameID].bets.length; j++){
        if (j==games[_gameID].outcome){
            games[_gameID].bets[j].status = STATUS_WIN;
        }
        else {
            games[_gameID].bets[j].status = STATUS_LOSE;
        }
    }
}
```

# Структура пројекта

## ▼ ETHEREUM\_SMART\_CONTRACT\_LOTTERY

> ethereum\_smart\_contract\_lottery

### ▼ lottery

> \_\_pycache\_\_

### ▼ migrations

> \_\_pycache\_\_

🔗 \_\_init\_\_.py

### ▼ static\lottery

> images

# style.css

### ▼ templates\lottery

<> checkIndex.html

<> createLottery.html

<> index.html

<> joinLottery.html

<> listLotteries.html

<> lotteryInfo.html

<> statusPage.html

🔗 \_\_init\_\_.py

🔗 admin.py

🔗 apps.py

🔗 forms.py

🔗 models.py

💎 smart\_contract.sol

🔗 tests.py

🔗 urls.py

🔗 views.py

≡ db.sqlite3

🔗 manage.py



# Рад апликације

- Web апликација
- Локални blockchain – Ganache софтвер
- Django фрејмворк
- Сви подаци добијају се из локалног blockchain-а (не постоји база података апликације)
- Демонстрација рада апликације

# Ganache софтвер

<https://www.trufflesuite.com/ganache>

ACCOUNTS

BLOCKS

TRANSACTIONS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK  
0

GAS PRICE  
20000000000

GAS LIMIT  
6721975

NETWORK ID  
5777

RPC SERVER  
HTTP://127.0.0.1:7545

MINING STATUS  
AUTOMINING

MNEMONIC

thrive lonely fall master dilemma often olive scheme domain roast secret evil

HD PATH

m/44'/60'/0'/0/account\_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x283ab4e76B10F84A42253D71fD5d7024ed0B144d	100.00 ETH	0	0	
0x75A4c451a32D6C08eE455Fb50049214442017428	100.00 ETH	0	1	
0x536100Cc789288B4CA35F340e5d7F9e37Ed5004B	100.00 ETH	0	2	
0xf8d3A5973Bb155D9dFD65Fb714EC045fAAD82563	100.00 ETH	0	3	
0xc3eE2644124bAd159b365bF83EA0C95A8b802b30	100.00 ETH	0	4	
0xAf16E631401a4E4E402cc013d46C33421D59C0E1	100.00 ETH	0	5	
0x138bFAf1aCc221054c37ED8d26f2c6D859912c76	100.00 ETH	0	6	



# Локални blockchain

- Разлози коришћења
- Није потребно посједовање стварне количине Ether валуте
- Није потребно плаћати провизије (fees) за трансакције
- Практично коришћење
- Непостојање ризика од губљења новца у случају грешке

# Прелазак на стварни Ethereum blockchain

- Проналазак начина како ћемо остварити комуникацију са blockchain-ом. Један начин је нпр. коришћење mainnet-а (blockchain који уствари врши функционалност преносе дигиталне валуте од пошиљаоца до примаоца). Комуникација са њим може се остварити преко сајта <https://infura.io/> гдје се потребно регистровати се и покренути властити пројекат, а затим одабрати blockchain за повезивање (mainnet је само један од њих).
- Посједовање одређене количине Ether криптовалуте



# Закључак

- Разлози за повјерење учесника
- Утицај сваког корисника на генерисање исхода
- Провјерљиви резултати
- Гаранција аутентичности кода који се покреће (паметног уговора) због уградње у blockchain
- Децентрализација – Ethereum мрежа
- Недостаци пројекта

# Хвала на пажњи!