

Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Seminarski rad iz predmeta:
Programski prevodioci

Tema:
Pametani ugovori "Rulet"

Student:
Nikola Mitrevski 603/2017

Predmetni profesor:
dr Vladimir M. Milovanović

Kragujevac 2021

Sadržaj:

1. Uvod	2
2. Opis delova programa	3
3. Opis korišćenja aplikacije.....	5
4. Literatura	15

1. Uvod

- U ovom projektu za razvoj pametnog ugovora, korišćena je veb aplikacija „Remix IDE“, koja omogućava pisanje pametnog ugovora u programskom jeziku „Solidity“.
- Solidity je objektno orijentisan programski jezik za pisanje pametnih ugovora.
- Remix IDE ima module za testiranje, otklanjanje grešaka, primenu pametnih ugovora i još mnogo toga i dostupan je na sledećem linku: <https://remix.ethereum.org/>.
- Remix IDE nam omogućava konekciju sa veb aplikacijom „Metamask“, koja je takođe korišćena u ovom projektu, ali za obavljanje transakcija i dostupna je na sledećem linku: <https://metamask.io/>.

2. Opis delova programa

- U prvoj liniji koda(slika 1), nalazi se direktiva, koja govori, koja verzija kompajlera će se koristiti za trenutnu Solidity datoteku.

```
1 pragma solidity ^0.4.22;
```

Slika 1 Prikaz linije koda u kojoj se nalazi pragma direktiva

- U narednih nekoliko linija koda(slika 2), nalaze se deklaracije promenljivih.

```
4 mapping (address => uint256) public winnings;  
5 uint8[] payouts;  
6 uint8[] numberRange;  
7 uint public randomNumber;
```

Slika 2 Prikaz linija koda u kojoj se nalaze potrebne deklaracije

- Deklaracija promenljive „winnings“ je namenjena za čuvanje vrednosti dobitaka za svakog pobednika, gde je svaki dobitak vezan za adresu pobednika(mapiranje).
- Deklaracija promenljive „payouts“ je namenjena za čuvanje vrednosti kvota.
- Deklaracija promenljive „numberRange“ je namenjena za čuvanje vrednosti koje predstavljaju krajnji opseg unetog broja.
- Deklaracija promenljive „randomNumber“ je namenjena za čuvanje generisanog slučajnog broja.
- Zatim u narednih nekoliko linija koda(slika 3), nalaze se definicija i deklaracija strukture, koja je namenjena za čuvanje adrese igrača, tipa, podtipa i uloga opklade.

```
34 struct Bet {  
35     address player;  
36     uint8 betType;  
37     uint8 number;  
38     uint256 betAmount;  
39 }  
40 Bet[] public bets;
```

Slika 3 Prikaz linija koda u kojoj se nalaze definicija i deklaracija strukture „Bet“

- I na kraju u narednih nekoliko linija koda(slika 4) imamo definicije sledećih metoda: „constructor“, „numberOfBets“, „bet“, „spinWheel“ i „cashOut“.
- Specijalna metoda konstruktor se poziva automatski prilikom stvaranja pametnog ugovara i služi za inicijalizaciju promenljivih.

```
42 constructor() public {...}
```

Slika 4 Prikaz linije koda u kojoj nalazi specijalna metoda konstruktor

- Metoda „numberOfBets“(slika 5) služi za vraćanje ukupnog broja uplaćenih opklada.

```
47 function numberOfBets() public view returns(uint) {...}
```

Slika 5 Prikaz linije koda u kojoj se nalazi metoda „numberOfBets“

- Metoda „bet“(slika 6) služi za uplatu opklade.

```
53 function bet(uint8 number, uint8 betType) public payable {...}
```

Slika 6 Prikaz linije koda u kojoj se nalazi metoda „bet“

- Metoda „spinWheel“(slika 7) služi za pronalaženje pobjednika.

```
67 ,      function spinWheel() public {...}
```

Slika 7 Prikaz linije koda u kojoj se nalazi metoda „spinWheel“

- Metoda „cashOut“(slika 8) služi za isplatu pobjednika.

```
134 ,     function cashOut() public {...}
```

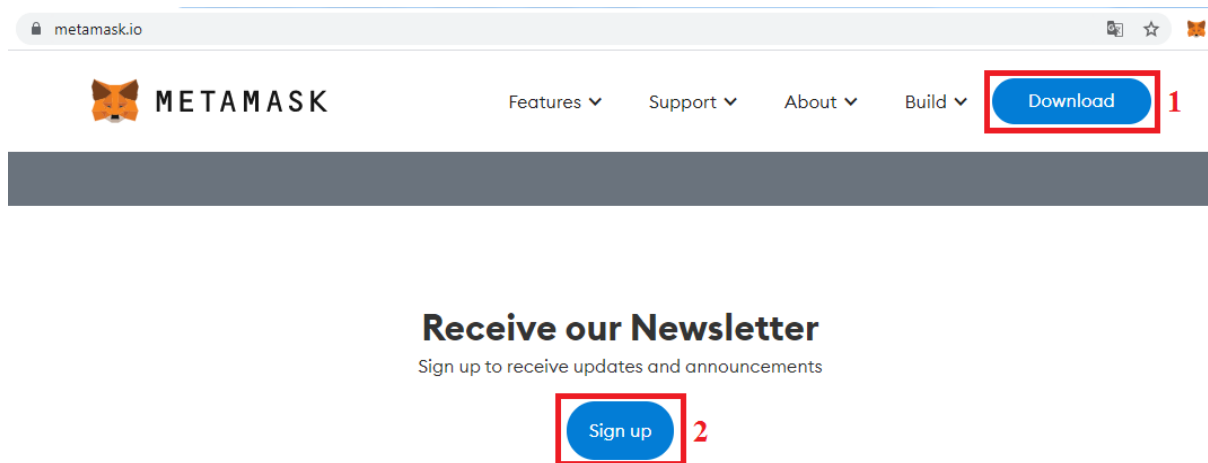
Slika 8 Prikaz linije koda u kojoj se nalazi metoda „cashOut“

3. Opis korišćenja aplikacije

- Da bi pametan ugovor bio uspešno primenjen, potrebno je pratiti sledeće korake:

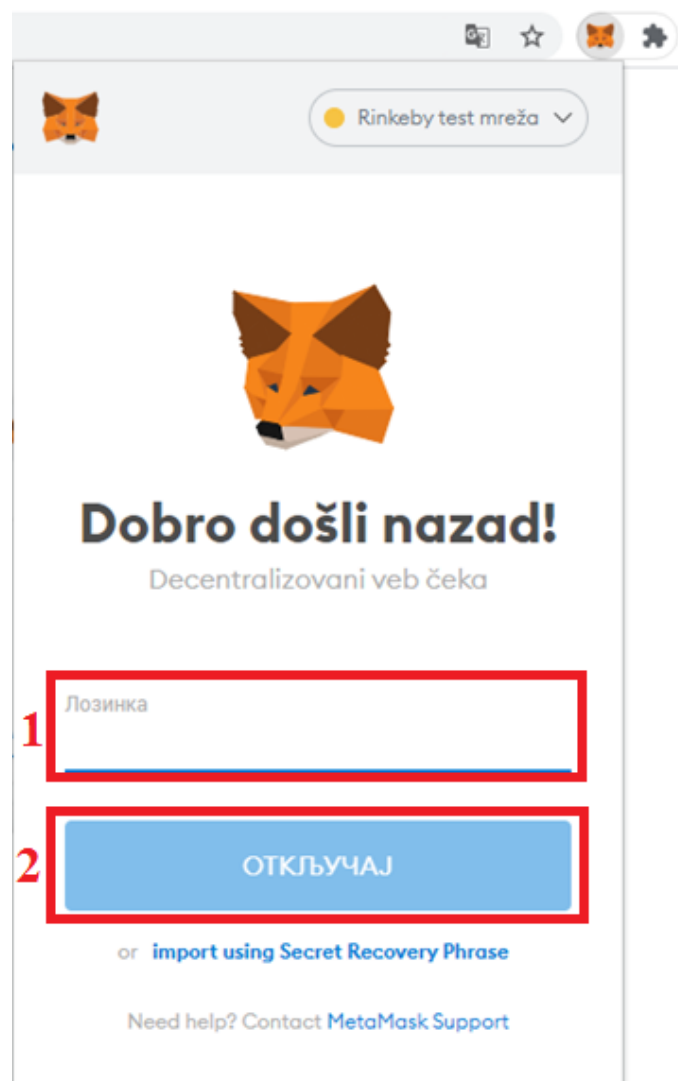
1. instaliranje Metamask proširenja za pretraživač(slika 9);

2. registrovanje na Metamask-u(slika 9);



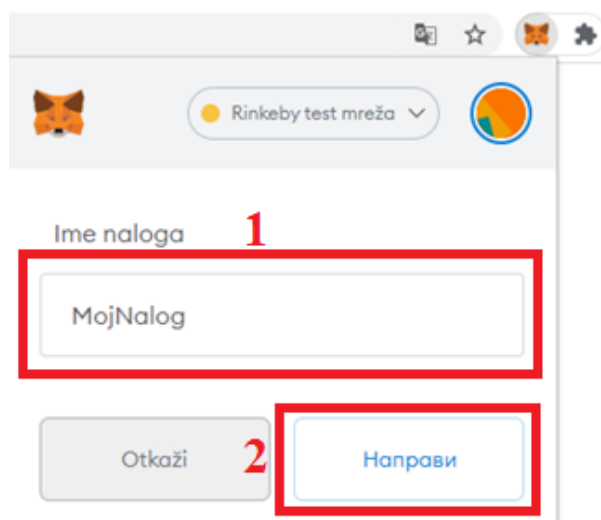
Slika 9 Demonstracija koraka „instaliranje i registrovanje na Metamask veb sajtu“

3. prijavljivanje na Mesamask-u(slika 10);



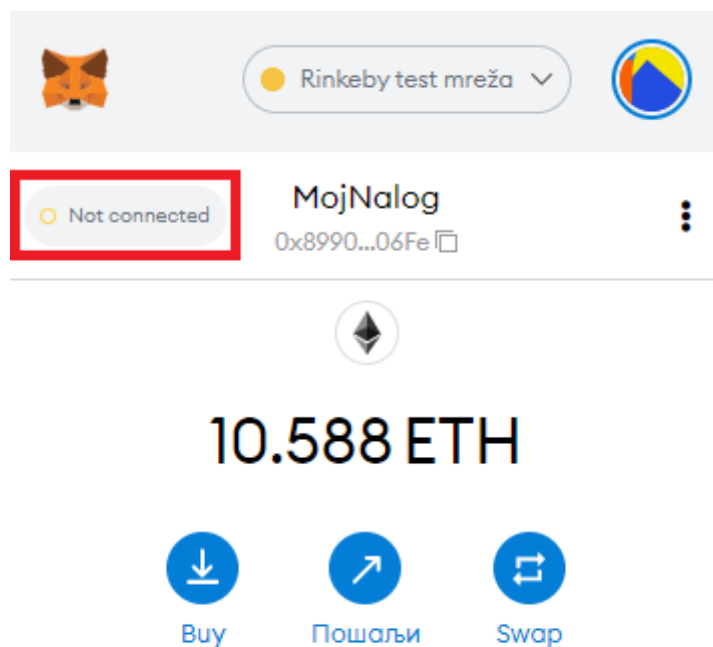
Slika 10 Demonstracija koraka „prijavljivanje na Metamask veb sajtu“

4. kreiranje kripto novčanika na Metamask-u(slika 11);



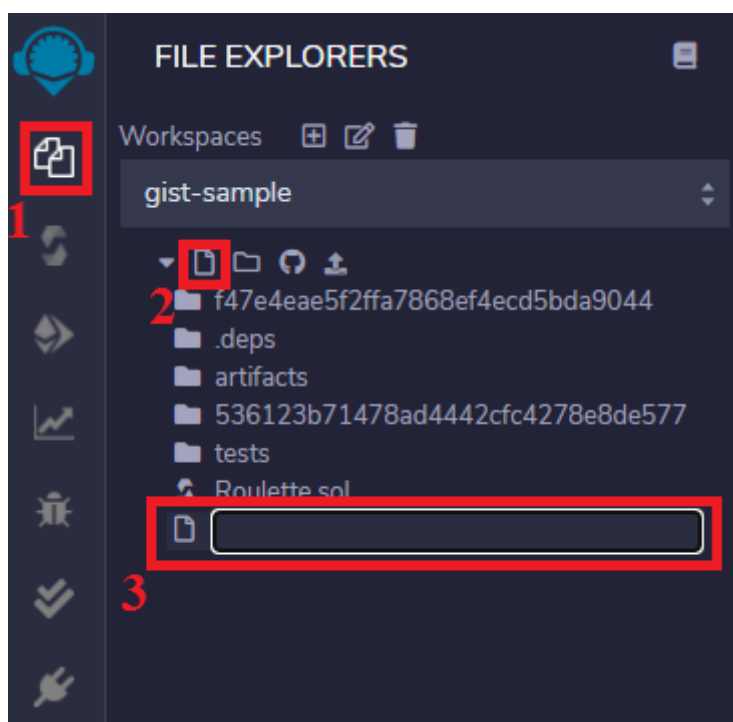
Slika 11 Demonstracija koraka „kreiranje kriptonovčanika na Metamask veb sajtu“

5. pristupanje veb sajtu: <https://faucet.rinkeby.io/> za dobijanje test novčica(Rinkeby) na kriptonovčaniku;
6. pristupanje veb sajtu: <https://remix.ethereum.org/>;
7. konekcija veb sajta Remix sa kriptonovčanikom iz Metamask-a(slika 12);



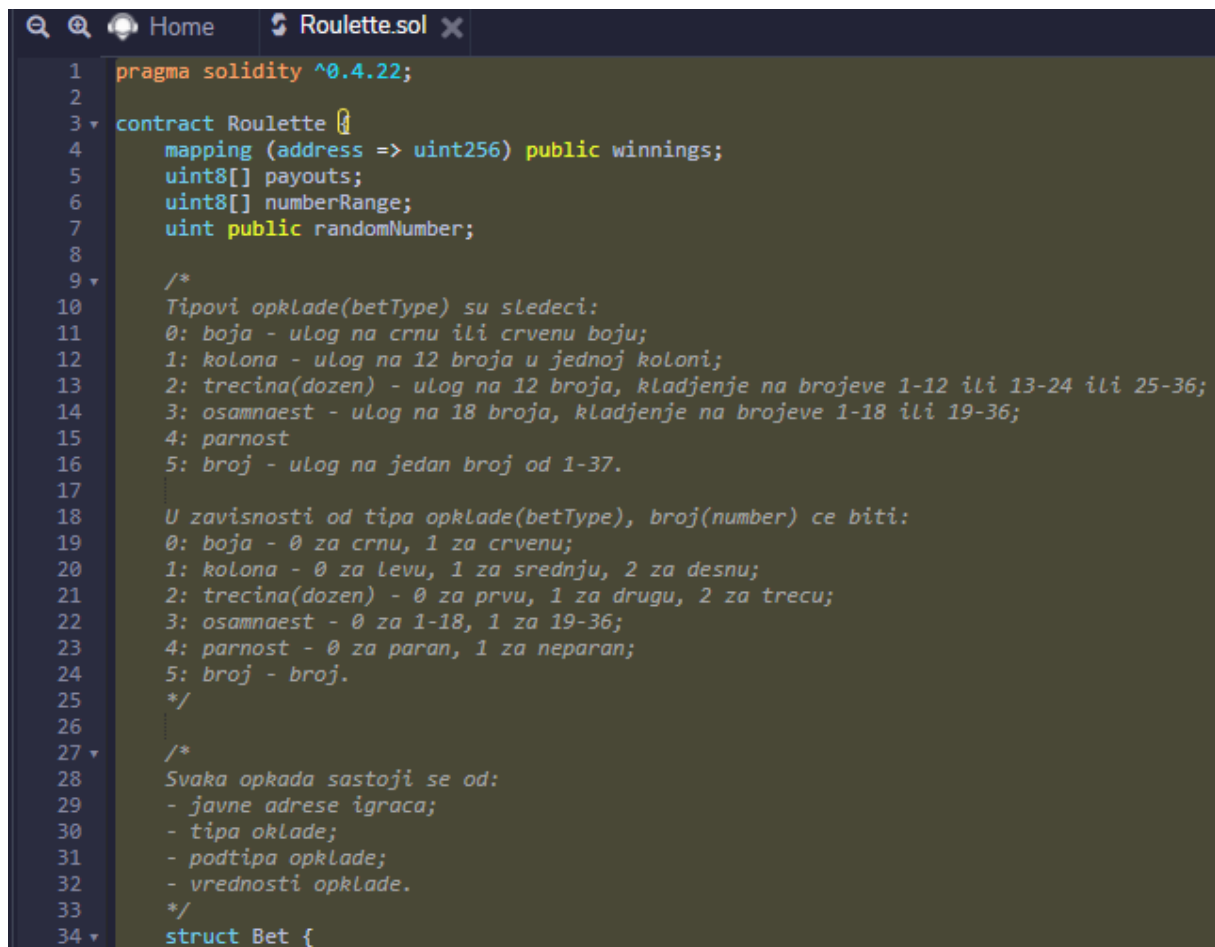
Slika 12 Demonstracija koraka „konekcija veb sajta Remix sa kriptonovčanikom iz Metamask veb sajta“

8. kreiranje Solidity datoteke(slika 13);



Slika 13 Demonstracija koraka „kreiranje Solidity datoteke na Remix veb sajtu“

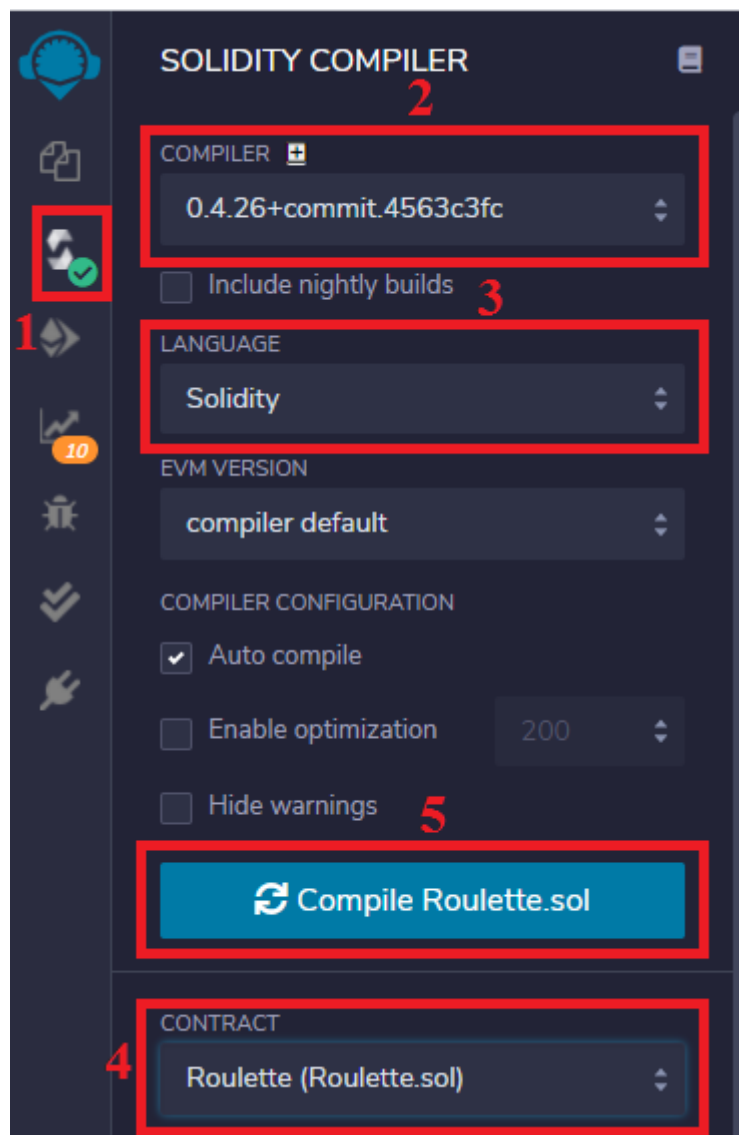
9. dodavanje koda u Solidity datoteku(slika 14);



```
1  pragma solidity ^0.4.22;
2
3  contract Roulette {
4      mapping (address => uint256) public winnings;
5      uint8[] payouts;
6      uint8[] numberRange;
7      uint public randomNumber;
8
9      /*
10     Tipovi opklade(betType) su sledeci:
11     0: boja - ulog na crnu ili crvenu boju;
12     1: kolona - ulog na 12 broja u jednoj koloni;
13     2: trecina(dozen) - ulog na 12 broja, kladjenje na brojeve 1-12 ili 13-24 ili 25-36;
14     3: osamnaest - ulog na 18 broja, kladjenje na brojeve 1-18 ili 19-36;
15     4: parnost
16     5: broj - ulog na jedan broj od 1-37.
17
18     U zavisnosti od tipa opklade(betType), broj(number) ce biti:
19     0: boja - 0 za crnu, 1 za crvenu;
20     1: kolona - 0 za levu, 1 za srednju, 2 za desnu;
21     2: trecina(dozen) - 0 za prvu, 1 za drugu, 2 za trecu;
22     3: osamnaest - 0 za 1-18, 1 za 19-36;
23     4: parnost - 0 za paran, 1 za neparan;
24     5: broj - broj.
25     */
26
27     /*
28     Svaka opkada sastoji se od:
29     - javne adrese igraca;
30     - tipa oklade;
31     - podtipa opklade;
32     - vrednosti opklade.
33     */
34     struct Bet {
```

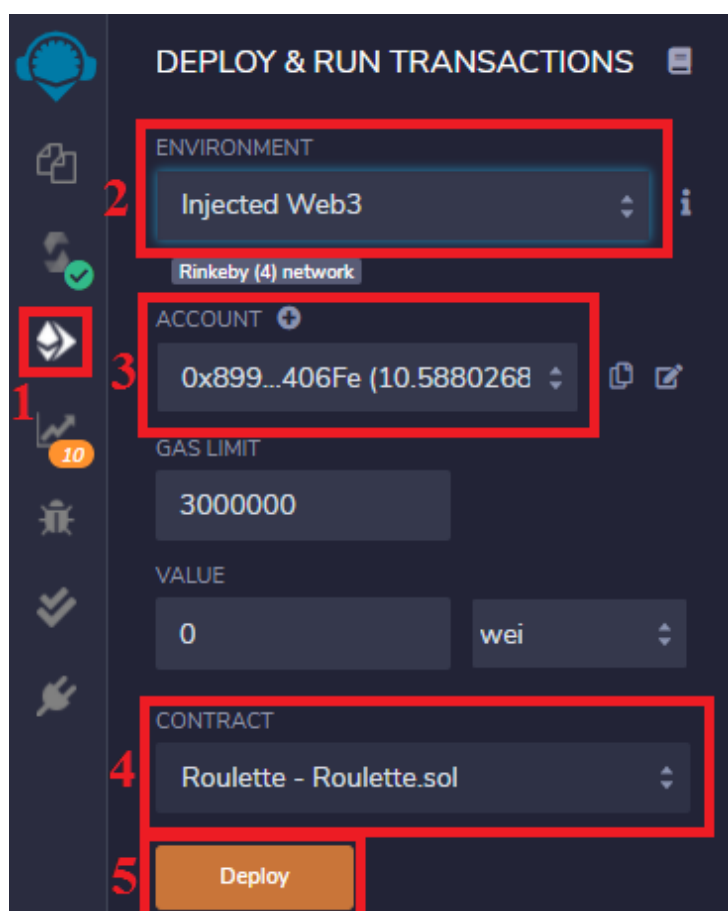
Slika 14 Demonstracija koraka „dodavanje Solidity koda u Solidity datoteku na Remix veb sajtu“

10. kompajliranje pametnog ugovora(slika 15);



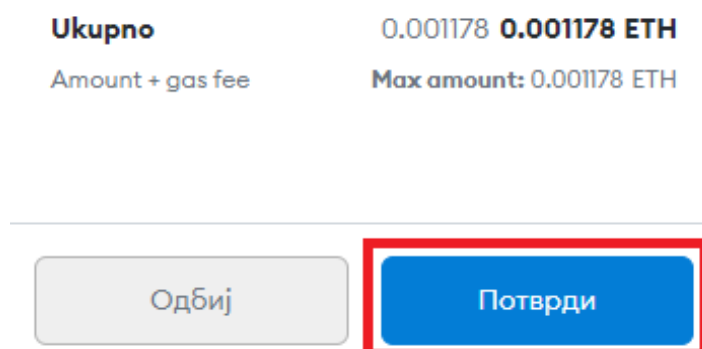
Slika 15 Demonstracija koraka „kompajliranje Solidity datoteke na Remix veb sajtu“

11. izgradnja pametnog ugovora(slika 16);



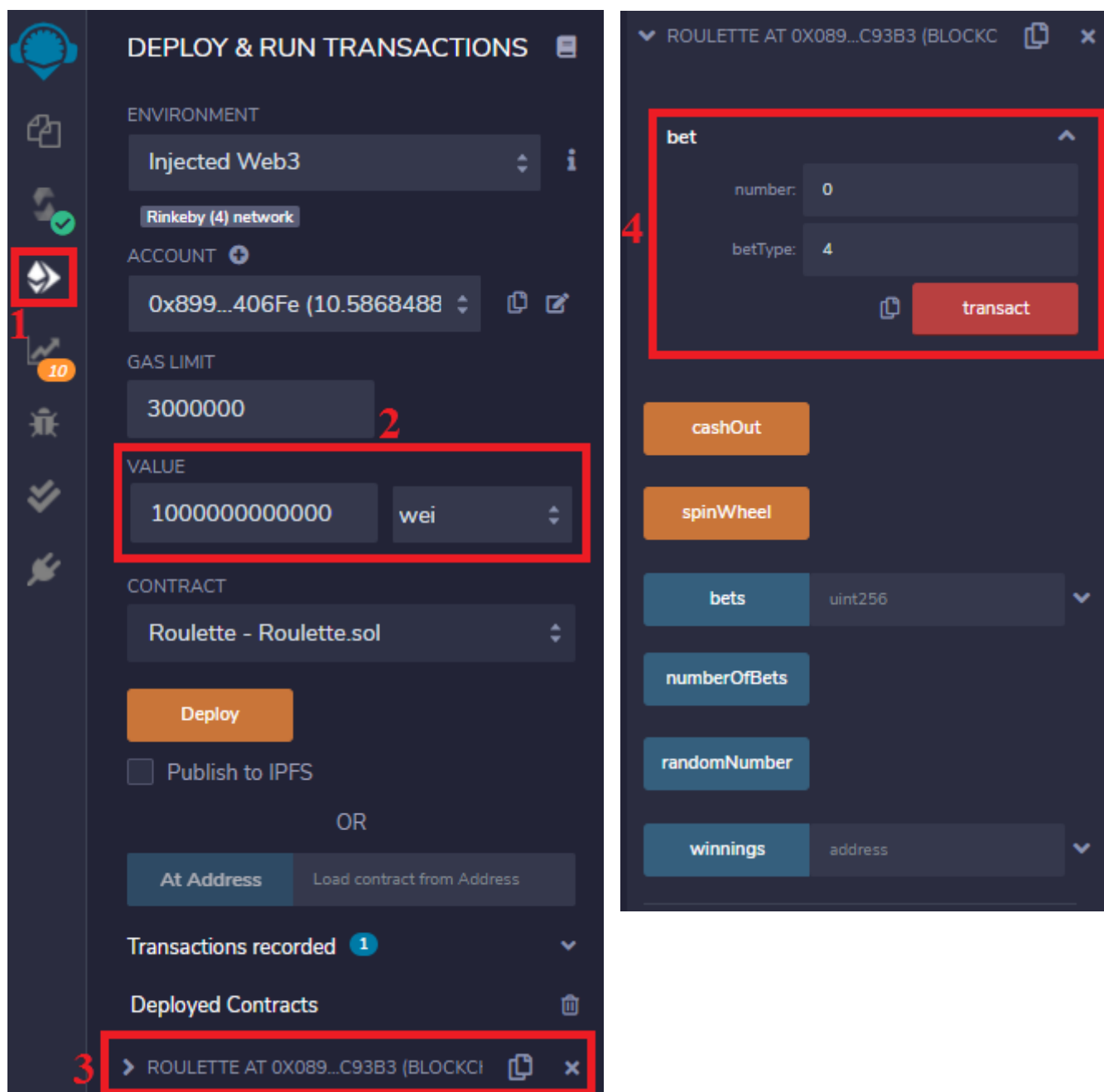
Slika 16 Demonstracija koraka „izgradnja Solidity datoteke na Remix veb sajtu“

12. potvrđivanje transakcije za izgradnju pametnog ugovora(slika 17);



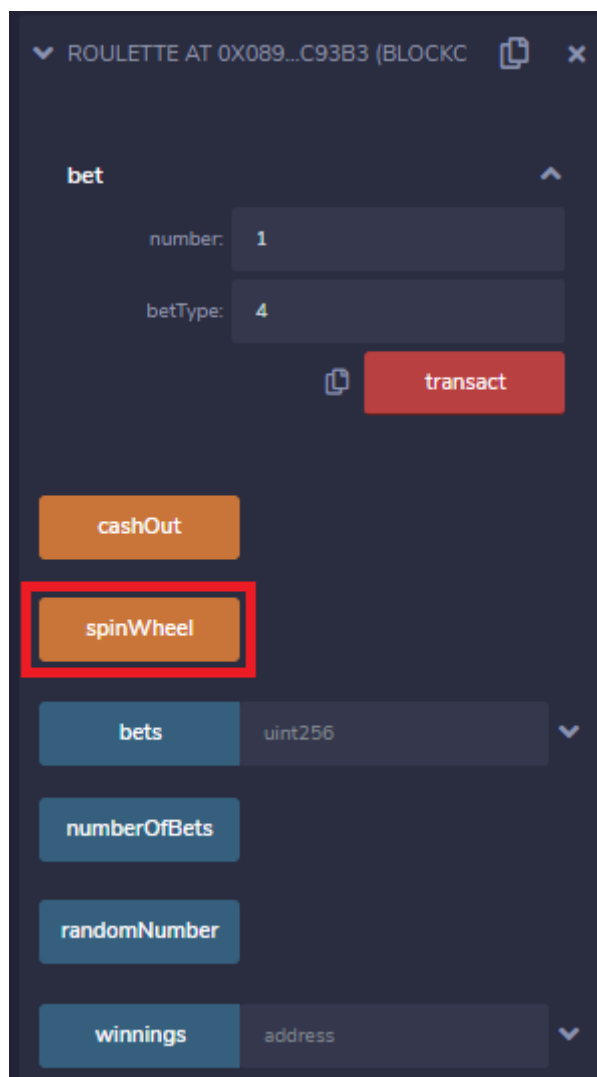
Slika 17 Demonstracija koraka „potvrđivanje transakcije za izgradnju pametnog ugovora na kriptonovčaniku iz Metamask veb sajta“

13. zadavanje vrednosti od minimum 0.01eth, popunjavanje opklade i izvršavanje transakcije(slika 18);



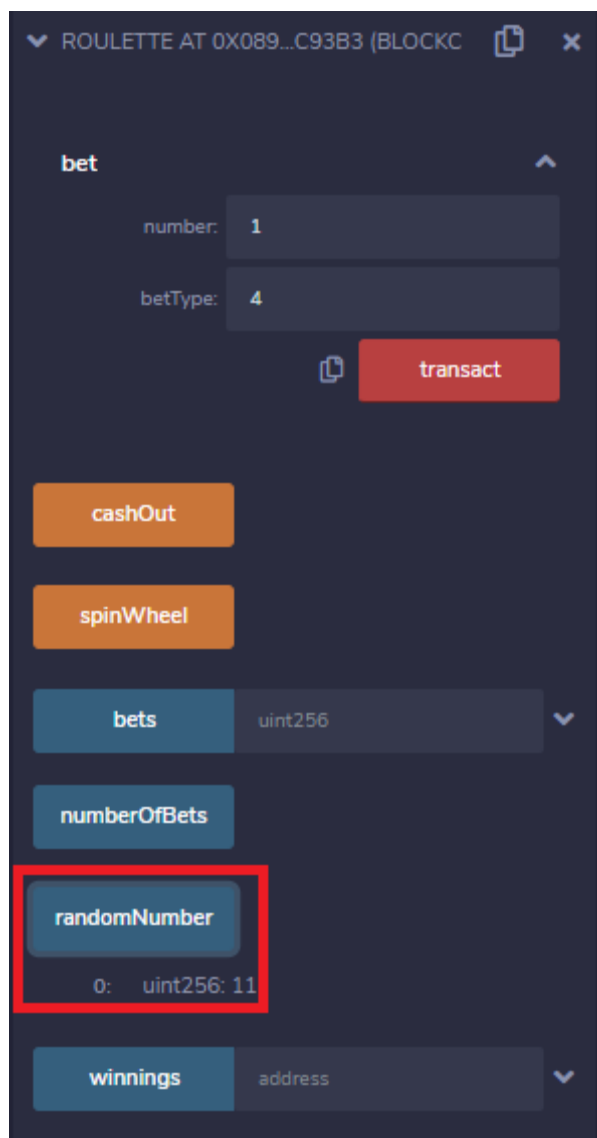
Slika 18 Demonstracija koraka „zadavanje vrednosti od 0.01eth, popunjavanje opklade i izvršavanje transakcije na Remix veb sajtu“

14. potvrđivanje transakcije za prihvatanje opklade;
15. određivanje pobjednika(slika 19);



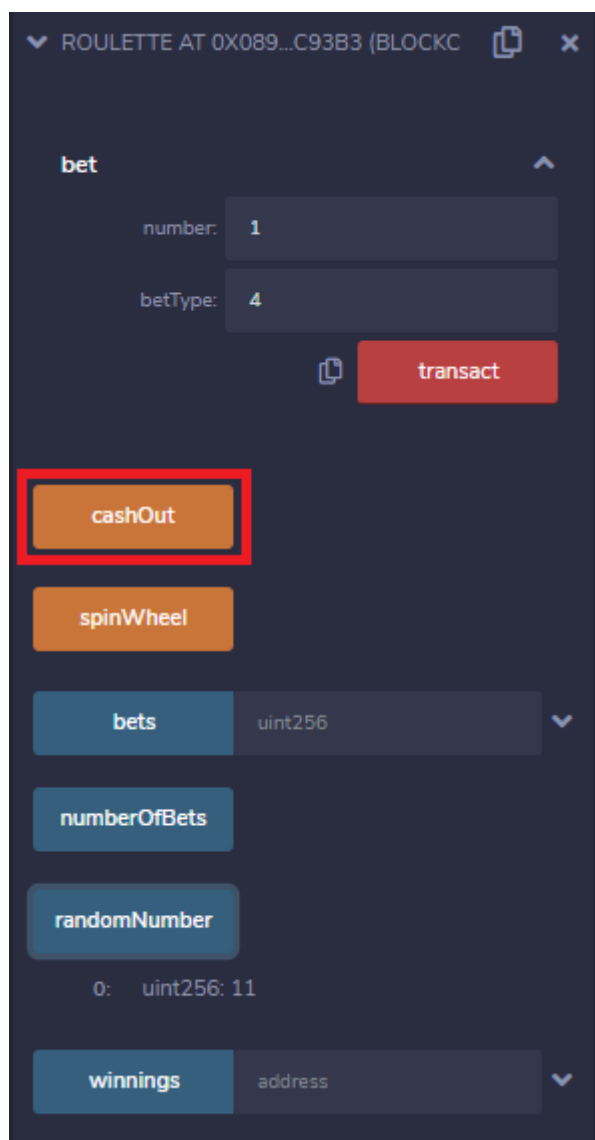
Slika 19 Demonstracija koraka „određivanje pobjednika“

16. potvrđivanje transakcije za određivanje pobjednika;
17. prikazivanje dobijenog slučajnog broja(slika 20);



Slika 20 Demonstracija koraka „prikazivanje dobijenog slučajnog broja“


18. isplata pobednika(slika 21);



Slika 21 Demonstracija koraka „isplata pobednika“

19. potvrđivanje transakcije za isplatu pobednika.

4. Literatura

- [1] Remix - Ethereum IDE: Creating and Deploying a Contract,
https://remix-ide.readthedocs.io/en/latest/create_deploy.html, 26.08.2021 (11:06);
- [2] YOUTUBE channel „howCode“:  Make your own lottery with Ethereum,
<https://www.youtube.com/watch?v=6GFOwXM69TQ>, 26.08.2021 (11:07).