Τελική εργασία blockchain

Νίκος Βασιλάκης Π17013, Παναγιώτης Ανδρέου Π17006

Ιούλιος 2024

Περιεχόμενα

1 Περιγραφή **3**

2 Υλοποίηση **4**

2.1 add_ballot(string memory q) 4
2.2 select_ballot(uint s) 4
2.3 close_vote() 5
2.4 vote_selected(bool c) 5
2.5 show_votes() 5
2.6 ballots 5

3 Εκτέλεση **7**

4 Επίλογος **23**

1 Περιγραφή

Ο σκοπός της εργασίας είναι να δημιουργηθεί ένα πρόγραμμα με βλοκσηαιν , το οποίο μπορεί κάποιος να ψηφίζει και να δημιουργηθεί ψηφοδέλτια μέσο ενός συμβολαίου. Στο πρόγραμμα θα χρησιμοποιήσουμε αυτές τις ενέργειες: Δημιουργία ψηφοδελτίου (μία ερώτηση για απάντηση) Κλείσιμο ψηφοδελτίου (Για να μην παίρνει άλλες ψήφους) Ψήφισμα. Εμφάνιση αποτελεσμάτων (με τον οποιοδήποτε τρόπο).

2 Υλοποίηση

Το πρόγραμμα έχει φτιαχτεί σε solidity (.sol αρχείο) και για να τρέξει μπορείτε να αντιγράψετε τον κώδικα , ανοίγοντας τον με Notepad , ή με το να εισάγετε το πρόγραμμα στο site.

Αρχικά θα χρειαστεί να δημιουργηθεί η δημόσια (public) θέση αποθήκευσης των αποτελεσμάτων. Για να γίνει αυτό φτιάξαμε ένα mapping με στρυστ αντικείμενα και κλειδί έναν αριθμό από το 1 έως το ποσό των ψηφοδελτίων. Επειδή στο mapping δεν μπορούμε να διαβάσουμε όλες τις τιμές που υπάρχουν , φτιάξαμε μία μεταβλητή με το πόσα ψηφοδέλτια υπάρχουν και θα μπορούμε να τα διαβάζουμε από το 1 μέχρι την τιμή αυτή. Μέσα στο στρυστ θα είναι το ψηφοδέλτιο με την ερώτηση μαζί με τις διευθύνσεις των χρηστών που απάντησαν ναι ή όχι στην ερώτησή μέσω 2 πινάκων και μια μεταβλητή που μας λέει αν το ψηφοδέλτιο είναι ανοιχτό.

Έπειτα υπάρχουν οι εξής συναρτήσεις με τις λειτουργείες τους.

2.1 add_ballot(string memory q)

Δημιουργεία ενός νέου ψηφοδελτίου με την ερώτηση που του δώσαμε

2.2 select_ballot(uint s)

Επειδή δεν υπάρχει ένας πιο άμεσος τρόπο για την επιλογή ενός ψηφοδέλτιού , για τις συναρτήσεις 2.3 και 2.4 θα πρέπει πρώτα να επιλέξουμε την ερώτηση με αυτήν την συνάρτηση , ως επιστροφή η συνάρτηση επιστρέφει το όνομα τις ερώτησης που διαλέξαμε. Η επιλογή είναι από 1 έως το ποσό των

ερωτήσεων που υπάρχουν.

2.3 close_vote()

Αλλάζει το is_opensε falseκαι με αυτόν τον τρόπο κλείνει η ψηφοφορία για την ερώτηση που διαλέξαμε

2.4 vote_selected(bool c)

Ψηφίζουμε την συγκεκριμένη ερώτηση. Η ψήφος είναι μοναδική για κάθε χρήστη για αυτό θα πρέπει ο κάθε χρήστης να μπορεί να ψηφίσει μόνο μια φορά σε κάθε ερώτηση. Μέσω των πινάκων στη struct μεταβλητή ξέρουμε ποιος χρήστης ψήφισε ναι ή όχι μέσω του αδδρεσς του. Άρα πρώτα πρέπει να βρούμε αν υπάρχει ο χρήστης με αυτό το address ήδη σε έναν από τους 2 πίνακες. Αν βρεθεί η ψήφος ακυρώνεται και του εμφανίζεται μήνυμα σφάλματος , διαφορετικά το αδδρεσς του πάει στο ανάλογο πίνακα ανάλογα του τι απάντησε. Για να ψηφίσει ναι θα πρέπει να εισαχθεί true (ή 1) για ναι ή false (ή 0) για όχι.

2.5 show_votes()

Μέσω τον struct που δημιουργήσαμε στο mappingμας , καλούμε ένα for loop και αποθηκεύουμε σε 3 προσωρινούς πίνακες την ερώτηση και ακριβώς από κάτω του τους 2 πίνακες με το πόσα ναι και όχι υπάρχουν (με αυτήν την σειρά) και τα εμφανίζει ως συνάρτηση view.

2.6 ballots

Επειδή το mapping ballot είναι public μπορούμε τα το εκτελέσουμε βάζοντας την τιμή (αρχίζοντας από το 0 όμως σαν

πίνακα) της θέσης τις ερώτησης και μας βγάζει ξεχωριστά αποτελέσματα για κάθε ερώτηση.

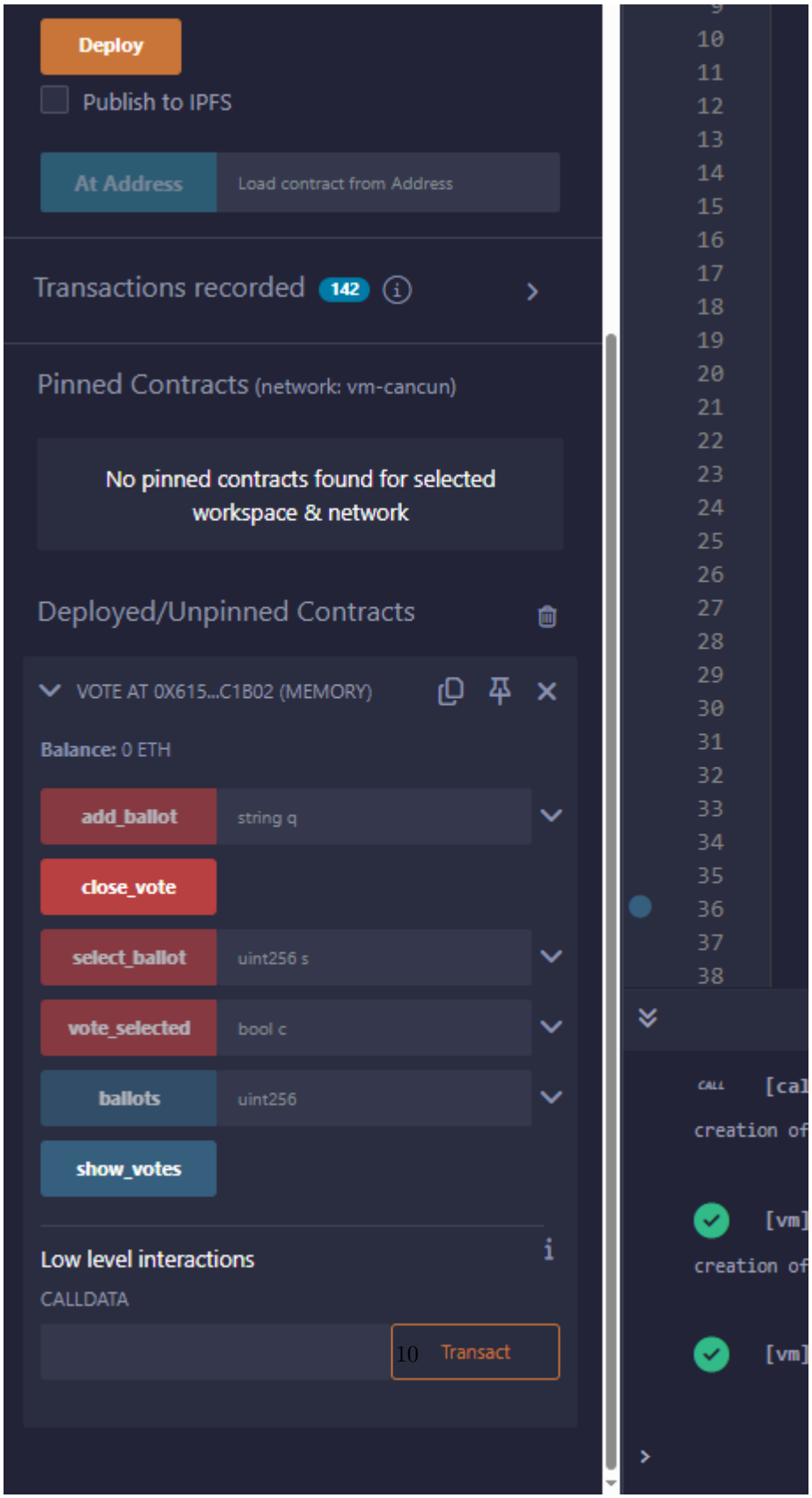
Έτσι με αυτόν τον τρόπο δημιουργήσαμε το πρόγραμμά μας για τα ψηφοδέλτια , το πρόβλημα είναι όμως ότι επειδή το εκτελούμε μόνοι μας , μπορεί να γίνει μόνο μία ψήφος σε κάθε ερώτηση.

3 Εκτέλεση

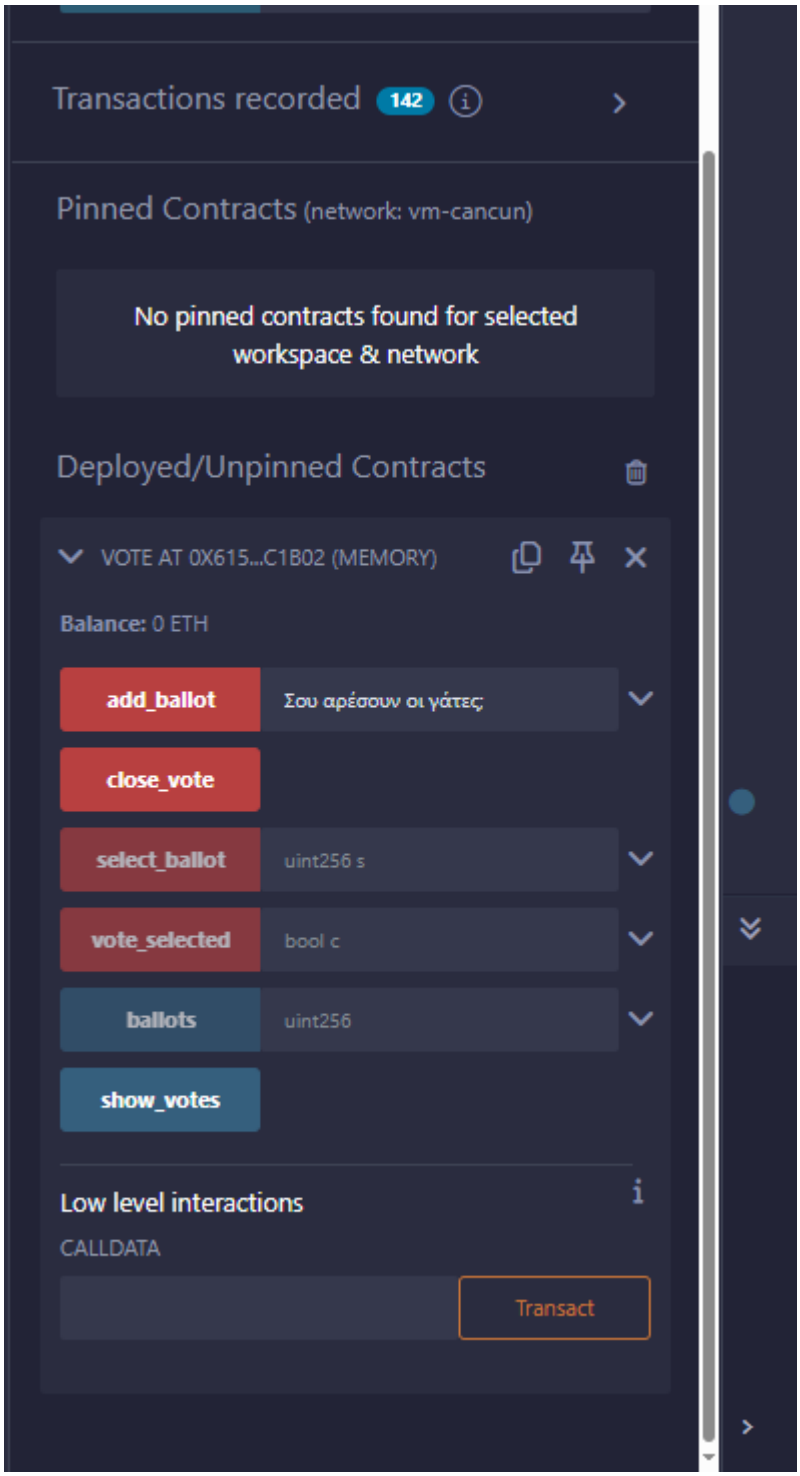
Για την εκτέλεση ενός παραδείγματος αρχικά ανοίγουμε το αρχείο μέσω του Remix IDE.


Home
123.sol
// SPDX-License-Identifier: MIT
pragma solidity >0.8.0 <9.0.0;
contract Vote {
struct Ballot{
string question;
bool is_open;
address[] yes;
address[] no;
}
Ballot b;
mapping (uint => Ballot) public ballots;
uint total_votes = 0;
uint selected;
function show_votes() public view returns(string[]
string[] memory q = new string[](total_votes);
uint[] memory y = new uint[](total_votes);
uint[] memory n = new uint[](total_votes);
for (uint i =0;i<total_votes;i++){
q[i]=ballots[i].question;
y[i]=ballots[i].yes.length;
n[i]=ballots[i].no.length;
}
return (q,y,n);
}
function select_ballot(uint s) public payable retur
require(s<=total_votes,"Ballot number doesn`t e
selected = s;
[vm] from: 0x5B3...eddC4 to: Vote.close_vote(uint256) 0xec7...F56
creation of Vote pending...

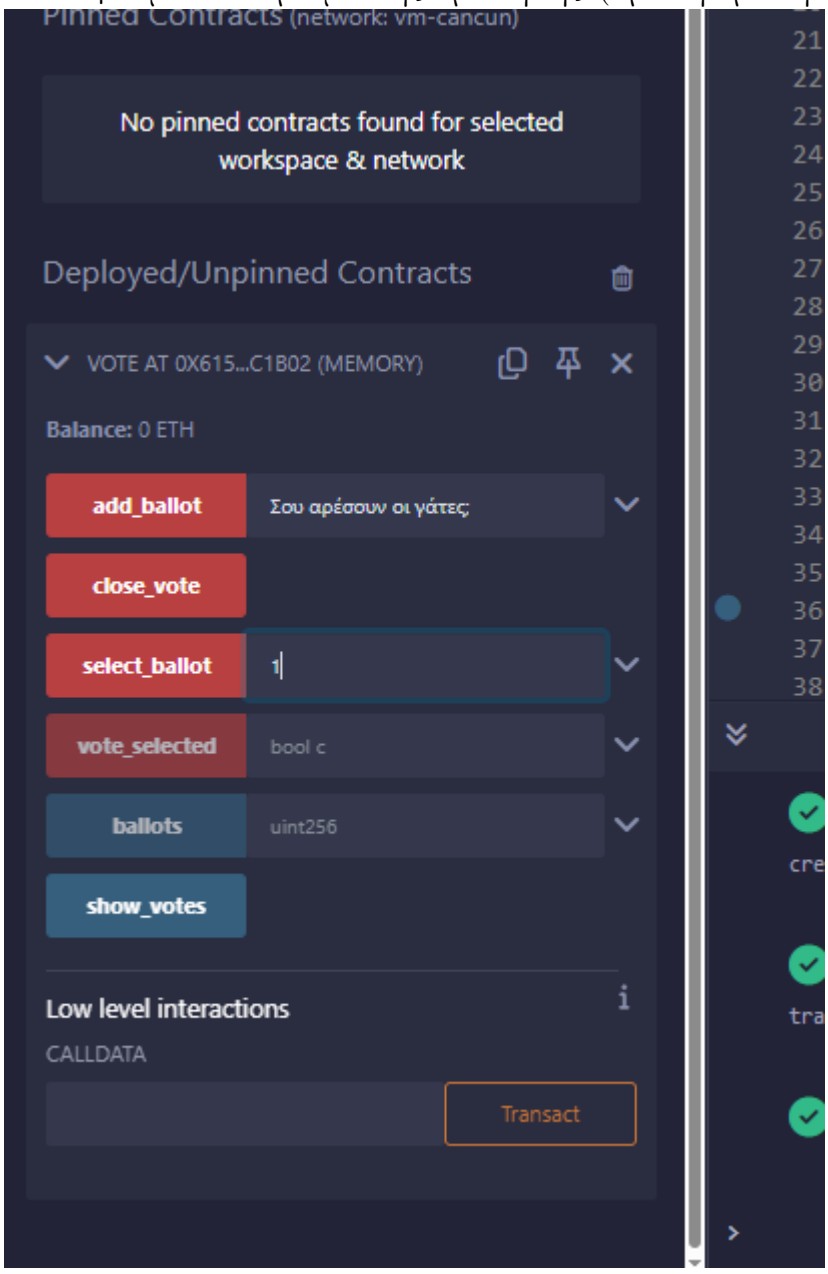
Κάνοντας ζομπιλε και δεπλοψ με custom 3000000 γας παίρνουμε αυτά τα αποτελέσματα:

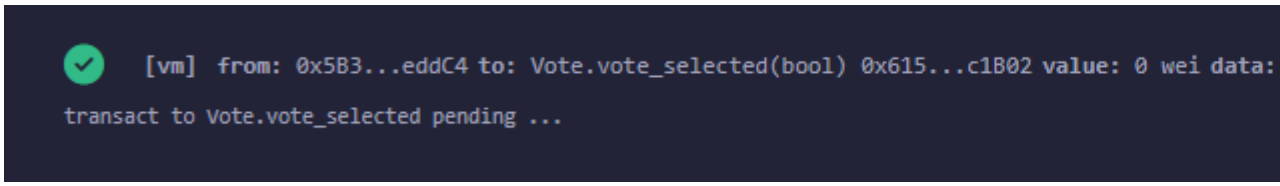

Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 142
Pinned Contracts (network: vm-cancun)
No pinned contracts found for selected workspace & network
Deployed/Unpinned Contracts
VOTE AT 0X615...C1B02 (MEMORY)
Balance: 0 ETH
add_ballot
string q
close_vote
select_ballot
uint256 s
vote_selected
bool c
ballots
uint256
show_votes
Low level interactions
CALLDATA
Transact
10
CALL [ca
creation of
[vm]
creation of
[vm]

Αρχικά κάνουμε μία ερώτηση:

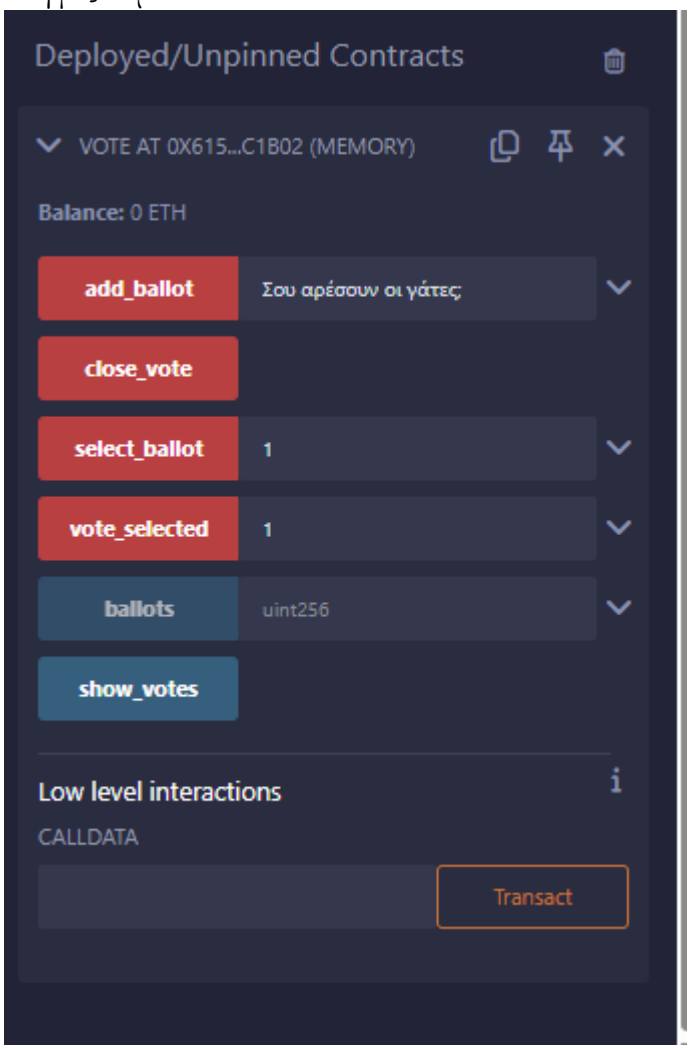

Transactions recorded 142
Pinned Contracts (network: vm-cancun)
No pinned contracts found for selected workspace & network
Deployed/Unpinned Contracts
VOTE AT 0X615...C1B02 (MEMORY)
Balance: 0 ETH
add_ballot
Σου αρέσουν οι γάτες;
close_vote
select_ballot
uint256 s
vote_selected
bool c
ballots
uint256
show_votes
Low level interactions
CALLDATA
Transact

Επιλέγουμε τον αριθμό της ερώτησης (πρώτη ερώτηση)



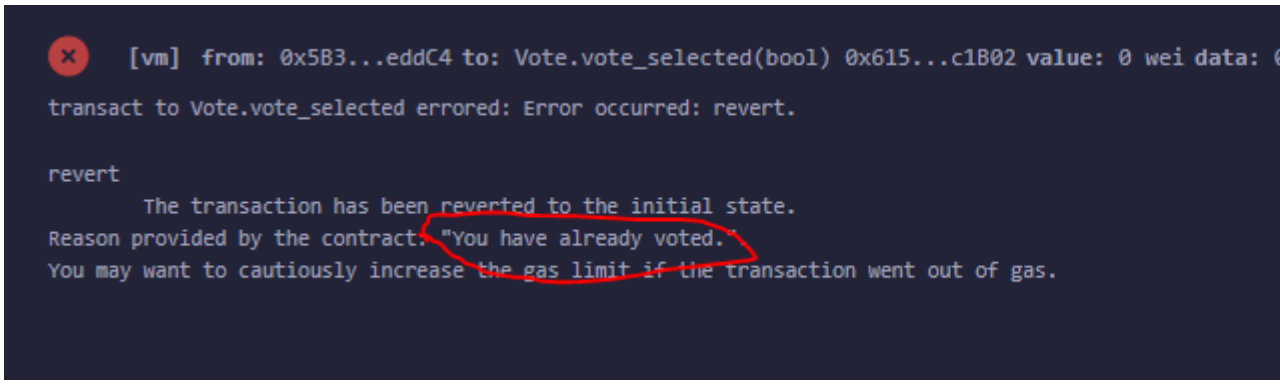


Ψηφίζουμε ναι.

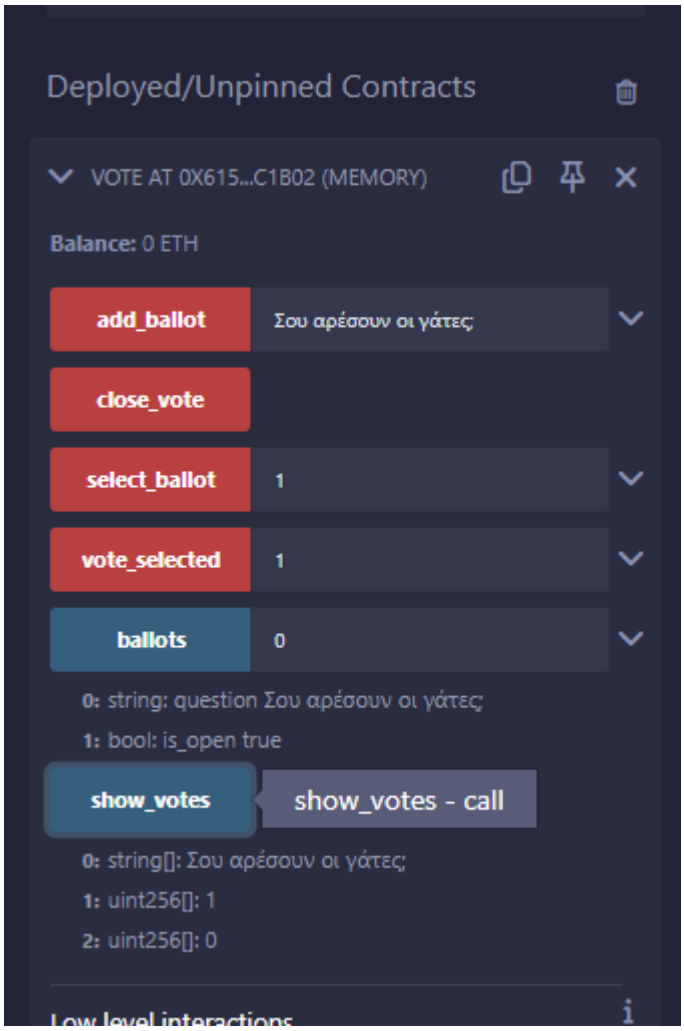


Αν πατήσουμε δεύτερη φορά να ψηφίσουμε:

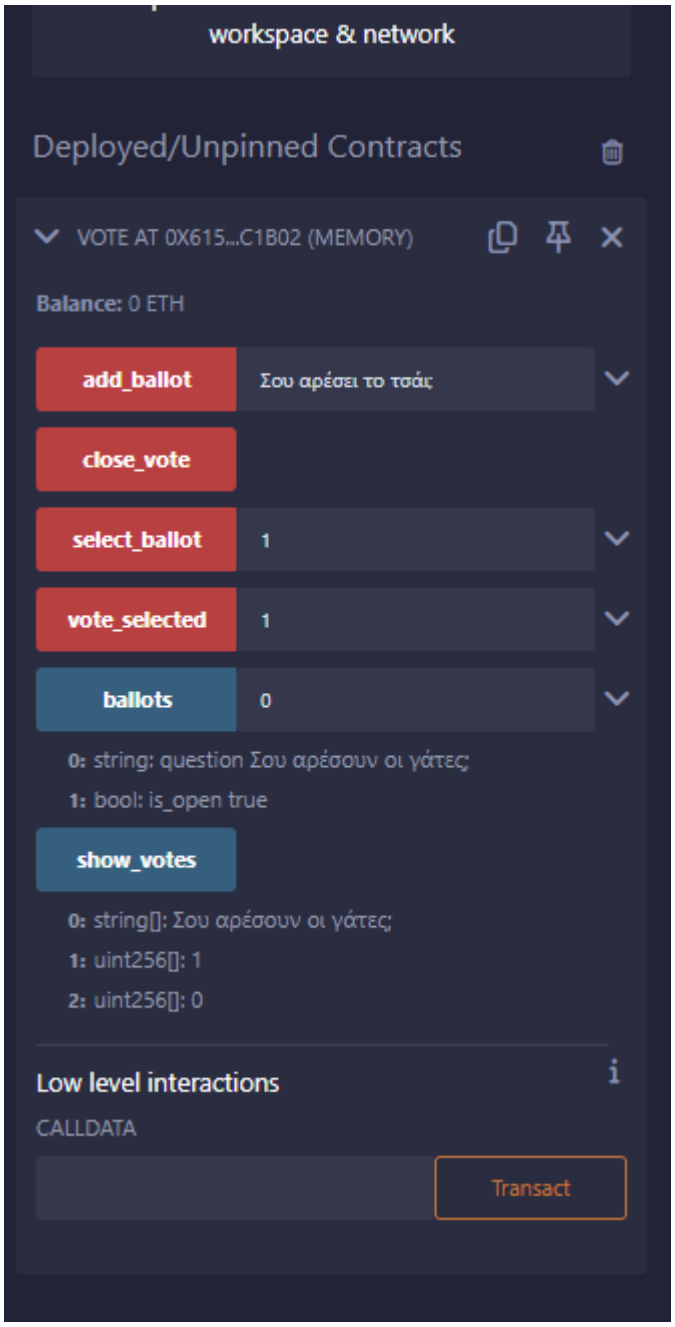
Μας λέει ότι ήδη ψηφίσαμε



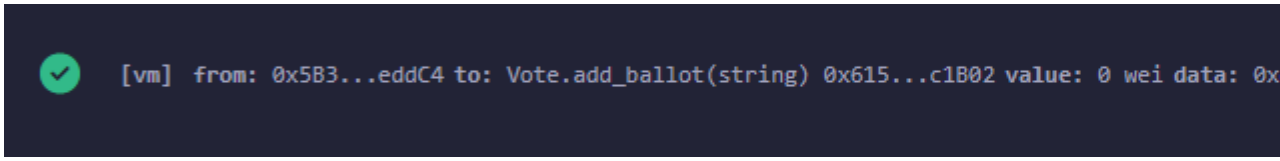
Πατώντας το ballots(με τιμή μηδέν γιατί είναι σαν πίνακας)
και το show_votes παίρνουμε αυτές τις τιμές



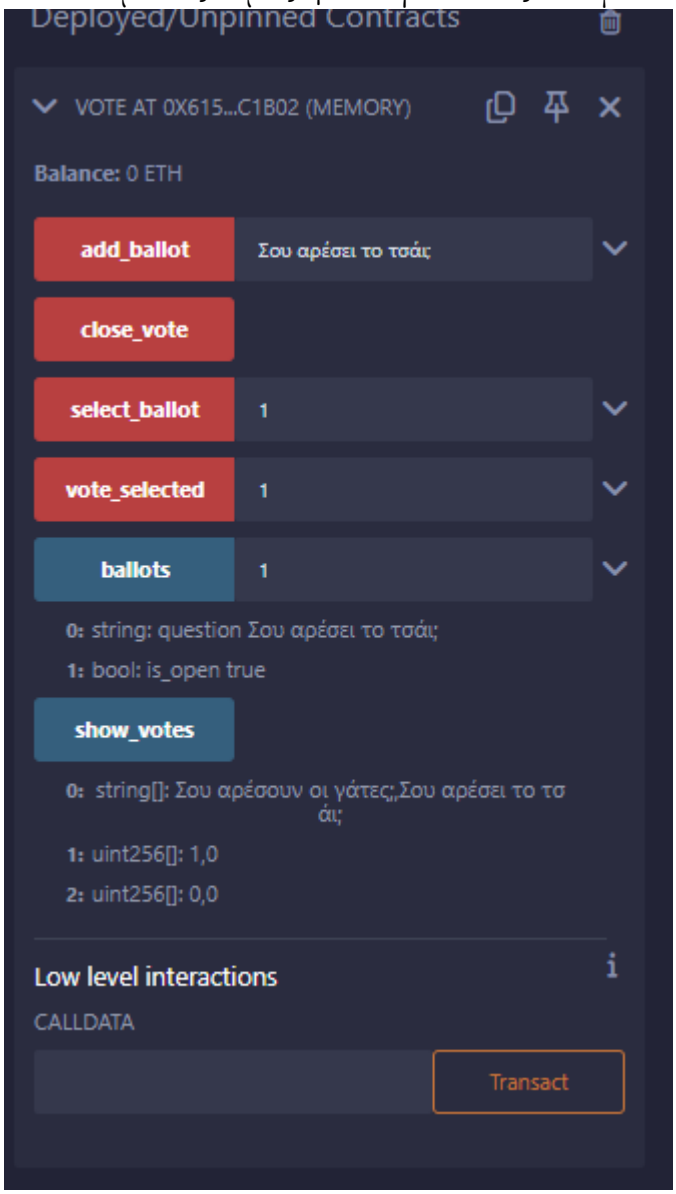
Κάνουμε μία νέα ερώτηση:


workspace & network
Deployed/Unpinned Contracts
VOTE AT 0X615...C1B02 (MEMORY)
Balance: 0 ETH
add_ballot
Σου αρέσει το τσάι;
close_vote
select_ballot
1
vote_selected
1
ballots
0
0: string: question Σου αρέσουν οι γάτες;
1: bool: is_open true
show_votes
0: string[]: Σου αρέσουν οι γάτες;
1: uint256[]: 1
2: uint256[]: 0
Low level interactions
CALLDATA
Transact

Έγινε δεκτό

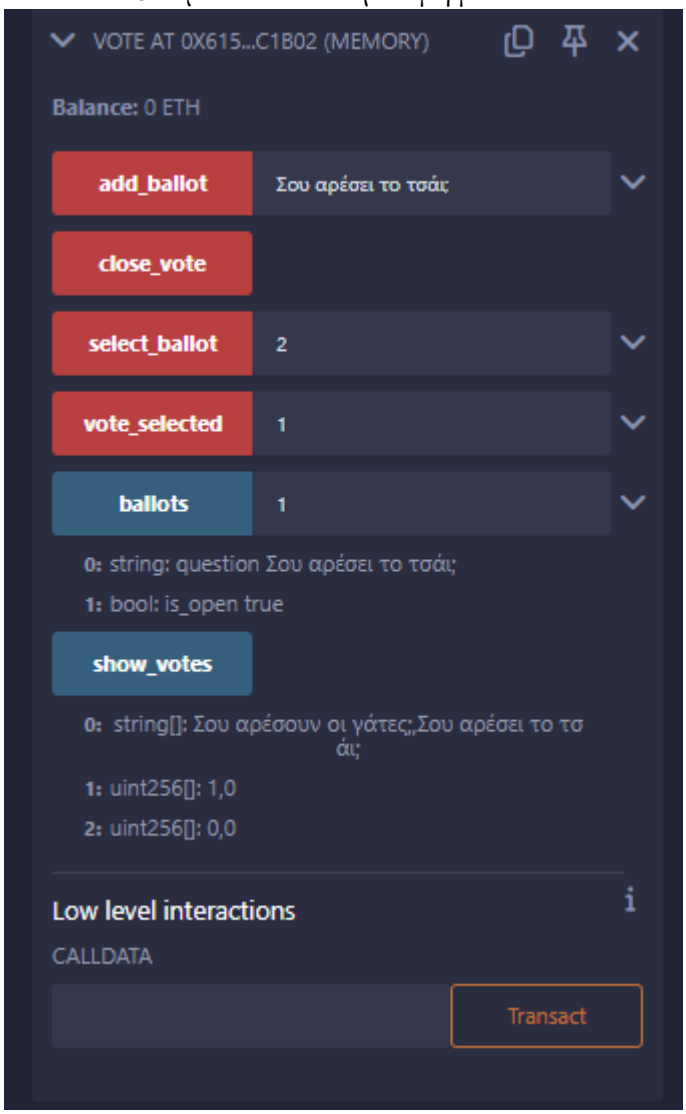


Βλέπουμε τις τιμές για σηοω_οτες και βαλλοτς (τιμή 1):

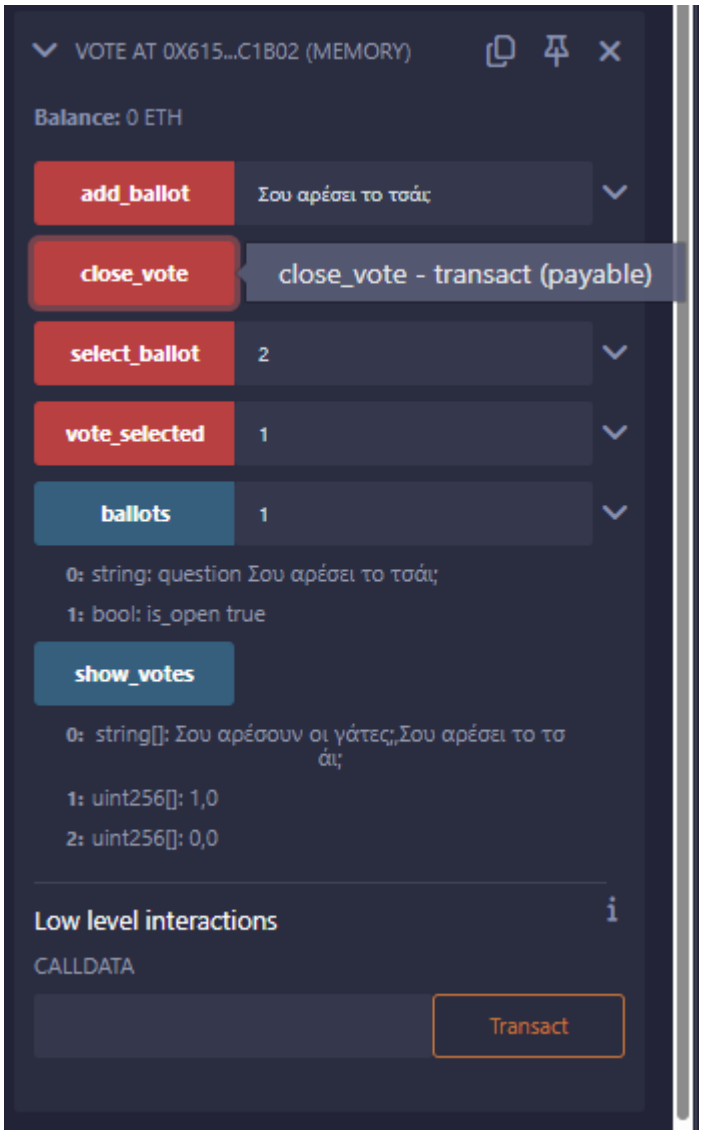


Έγινε με επιτυχία

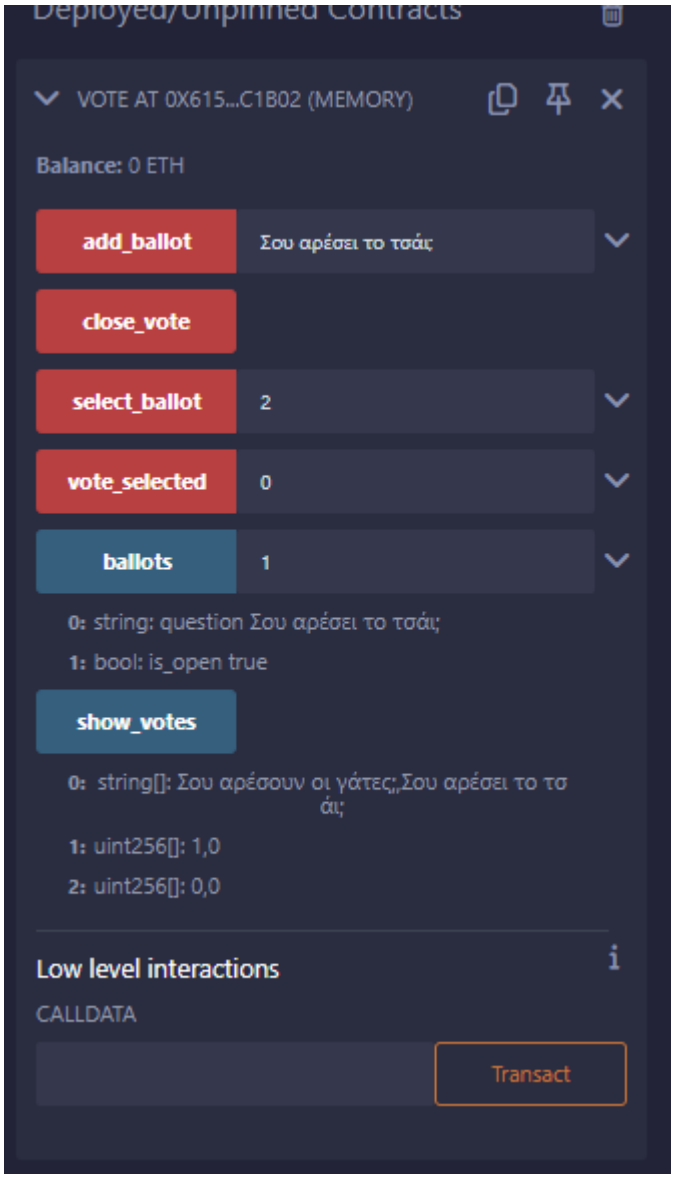
Επιλέξουμε το δεύτερο ψηφοδέλτιο

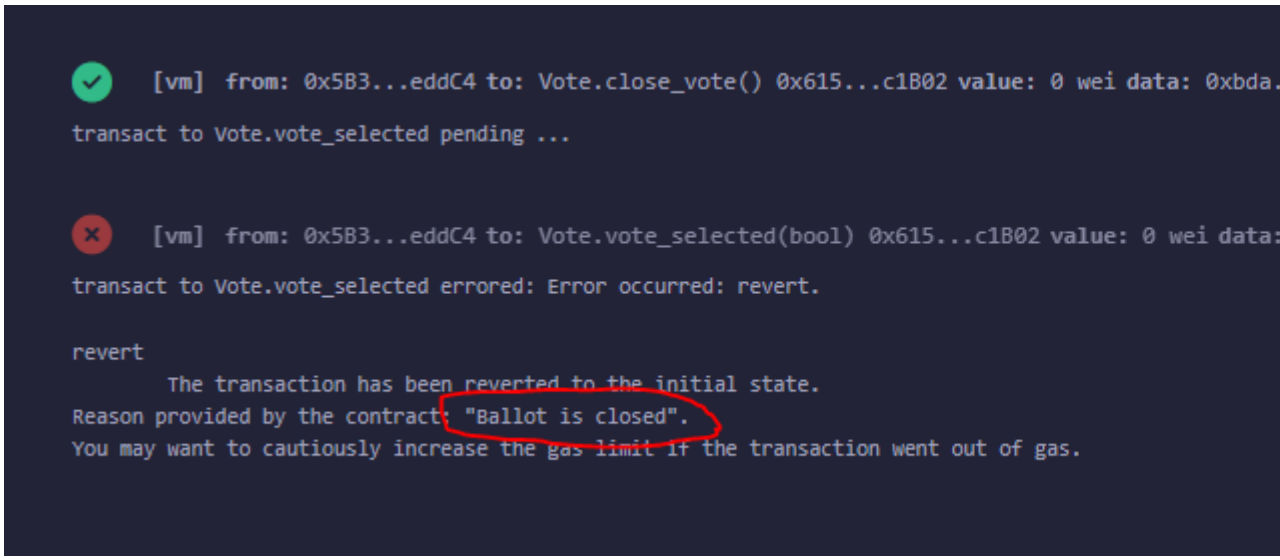


Και τώρα πάμε να κλείσουμε το ψηφοδέλτιο που επιλέξαμε



Άμα πάμε να ψηφίσουμε τώρα (πχ όχι)


Deployed/Unpinned Contracts
VOTE AT 0X615...C1B02 (MEMORY)
Balance: 0 ETH
add_ballot
Σου αρέσει το τσάι;
close_vote
select_ballot
2
vote_selected
0
ballots
1
0: string: question Σου αρέσει το τσάι;
1: bool: is_open true
show_votes
0: string[]: Σου αρέσουν οι γάτες;,Σου αρέσει το τσάι;
1: uint256[]: 1,0
2: uint256[]: 0,0
Low level interactions
CALLDATA
Transact



Μας λέει ότι έχει κλείσει το ψηφοδέλτιο

4 Επίλογος

Αυτή ήταν η άσκηση και αυτά ήταν τα ερωτήματα.
(Το κείμενο έχει δημιουργηθεί σε LaTeX)