

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**



**Μάθημα Προπτυχιακών Σπουδών:**  
Τεχνολογίες Blockchain και Εφαρμογές

**Εργασία Εξαμήνου**

Αντώνιος Ρούσσος

Ιούλιος 2024

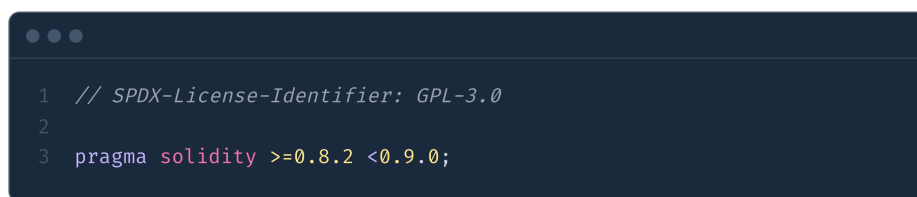
# Περιεχόμενα

<b>1</b>	<b>Ανάλυση Κώδικα</b>	<b>2</b>
1.1	Αρχική Ανάλυση . . . . .	2
1.2	Συνάρτηση Δημιουργίας Νέας Ψηφοφορίας . . . . .	2
1.3	Συνάρτηση Δημιουργίας Ψήφου σε μια Ψηφοφορία . . . . .	3
1.4	Συνάρτηση Κλεισίματος μιας Ψηφοφορίας . . . . .	3
1.5	Συνάρτηση Προβολής Αποτελεσμάτων . . . . .	3
<b>2</b>	<b>Στιγμιότυπα Οθόνης</b>	<b>6</b>
<b>3</b>	<b>Παράδειγμα Εκτέλεσης</b>	<b>7</b>
3.1	Δημιουργία Νέας Ψηφοφορίας . . . . .	7
3.2	Δημιουργία Ψήφου σε Ψηφοφορία . . . . .	7
3.3	Κλείσιμο Ψηφοφορίας . . . . .	7
3.4	Προβολή Αποτελεσμάτων . . . . .	7

# 1 Ανάλυση Κώδικα

## 1.1 Αρχική Ανάλυση

Για την ανάπτυξη του έξυπνου συμβολαίου (Smart Contract) χρησιμοποίησα τη γλώσσα προγραμματισμού Solidity και το διαδικτυακό περιβάλλον ανάπτυξης έξυπνων συμβολαίων Remix. Το Remix παρέχει έναν εύκολο τρόπο να τρέξει κάποιος ένα έξυπνο συμβόλαιο σε ένα εικονικό δίκτυο Ethereum (testnet). Στο αρχείο του έξυπνου συμβολαίου, αρχικά, αναφέρεται ο αναγνωριστικός κωδικός άδειας, ο οποίος καθορίζει την άδεια υπό την οποία διατίθεται ο κώδικας. Επέλεξα να χρησιμοποιήσω την άδεια GPL-3.0 η οποία δηλώνει πως ο κώδικας διατίθεται υπό την Άδεια Γενικής Δημόσιας Χρήσης. Έπειτα, αναγράφω την έκδοση του Solidity compiler όπως φαίνεται στην εικόνα 1.



```
1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity >=0.8.2 <0.9.0;
```

Εικόνα 1: Ορισμός έκδοσης του Solidity compiler.

Η δομή της ψηφοφορίας ορίζεται από ένα struct (βλ. εικόνα 2) με τις παρακάτω μεταβλητές κατάστασης:

- string **question**: για την αποθήκευση του ερωτήματος της ψηφοφορίας. Σημαντικό είναι πως το ερώτημα πρέπει να είναι κλειστού τύπου και να μπορεί να απαντάται μόνο με ναι ή όχι.
- bool **isOpen**: θα ορίζει αν η ψηφοφορία είναι ανοιχτή ή κλειστή.
- address **creator**: για την αποθήκευση της διεύθυνσης του δημιουργού της ψηφοφορίας.
- address[] **voters**: για την αποθήκευση των διευθύνσεων που έχουν υποβάλει ψήφο.
- mapping(address => bool) **votes**: για την αποθήκευση των ψήφων. Το κλειδί (key) του map είναι τύπου διεύθυνσης και η τιμή (value) τύπου boolean.
- mapping(address => bool) **hasVoted**: για την αποθήκευση των διευθύνσεων που έχουν ψηφίσει.

Οι ψηφοφορίες αυτές αποθηκεύονται σε έναν πίνακα (array).

## 1.2 Συνάρτηση Δημιουργίας Νέας Ψηφοφορίας

Η δημιουργία μιας νέας ψηφοφορίας πραγματοποιείται με τη συνάρτηση createVoting (βλ. εικόνα 3), η οποία δέχεται ως όρισμα την ερώτηση της ψηφοφορίας, φτιάχνει την ψηφοφορία και την αποθηκεύει στον πίνακα votingEvents. Για την απόκτηση της διεύθυνσης του καλούντος χρησιμοποιείται το προκαθορισμένο αντικείμενο msg, που παρέχει πληροφορίες για την τρέχουσα συναλλαγή.

```

1 struct VotingEvent {
2     string question;
3     bool isOpen;
4     address creator;
5     address[] voters;
6     mapping(address => bool) votes;
7     mapping(address => bool) hasVoted;
8 }

```

Εικόνα 2: Η δομή της ψηφοφορίας.

```

1 function createVoting(string calldata _question) public {
2     VotingEvent storage newVoting = votingEvents.push();
3     newVoting.creator = msg.sender;
4     newVoting.question = _question;
5     newVoting.isOpen = true;
6 }

```

Εικόνα 3: Συνάρτηση δημιουργίας μιας ψηφοφορίας.

### 1.3 Συνάρτηση Δημιουργίας Ψήφου σε μια Ψηφοφορία

Η δημιουργία ψήφου σε μια ψηφοφορία μπορεί να γίνει με τη κλήση της συνάρτησης `vote` (βλ. εικόνα 4), η οποία δέχεται ως ορίσματα τον δείκτη (index) της ψηφοφορίας βάσει του πίνακα και την ψήφο του χρήστη. Έπειτα, χρησιμοποιεί την εντολή `require` για να κάνει τους ακόλουθους ελέγχους πριν την εισαγωγή της ψηφοφορίας στο blockchain:

1. Έλεγχος εγκυρότητας δείκτη ψήφου
2. Έλεγχος αν η ψηφοφορία που αντιστοιχεί στον δείκτη είναι ανοιχτή

Γνωρίζοντας ότι οι παραπάνω έλεγχοι είναι έγκυροι, η εκτέλεση του προγράμματος συνεχίζεται με την εισαγωγή της ψήφου στην εκάστοτε ψηφοφορία.

### 1.4 Συνάρτηση Κλεισίματος μιας Ψηφοφορίας

Το κλείσιμο μιας ψηφοφορίας ορίζεται στη συνάρτηση `closeVoting` (βλ. εικόνα 5), η οποία δέχεται ένα όρισμα για τον δείκτη της ψηφοφορίας. Μια ψηφοφορία μπορεί να πάψει να δέχεται ψήφους μόνο από την διεύθυνση που είχε δημιουργηθεί. Αυτό πραγματοποιείται με έναν απλό έλεγχο πριν το κλείσιμο της ψηφοφορίας, ο οποίος ελέγχει αν η διεύθυνση του καλούντος είναι ίση με αυτή του δημιουργού της.

### 1.5 Συνάρτηση Προβολής Αποτελεσμάτων

Η συνάρτηση `getVotes` (βλ. εικόνα 6) ανακτά τις ψήφους από μια συγκεκριμένη ψηφοφορία και επιστρέφει δύο πίνακες, έναν με τις διευθύνσεις των ψηφοφόρων και

```

1 function vote(uint256 _index, bool _vote) public {
2     require(_index < votingEvents.length, "Invalid voting index");
3     require(votingEvents[_index].isOpen, "Voting is closed");
4
5     VotingEvent storage voting = votingEvents[_index];
6
7     // Check if the voter has already voted
8     if (!voting.hasVoted[msg.sender]) {
9         voting.voters.push(msg.sender);
10        voting.hasVoted[msg.sender] = true;
11    }
12
13    voting.votes[msg.sender] = _vote;
14 }

```

Εικόνα 4: Συνάρτηση ψηφίσματος σε ψηφοφορία.

```

1 function closeVoting(uint256 _index) public {
2     require(_index < votingEvents.length, "Invalid voting index");
3     require(votingEvents[_index].creator == msg.sender, "You are not
the creator of this voting");
4     require(votingEvents[_index].isOpen, "Voting is already closed");
5
6     votingEvents[_index].isOpen = false;
7 }

```

Εικόνα 5: Συνάρτηση κλεισίματος μιας ψηφοφορίας.

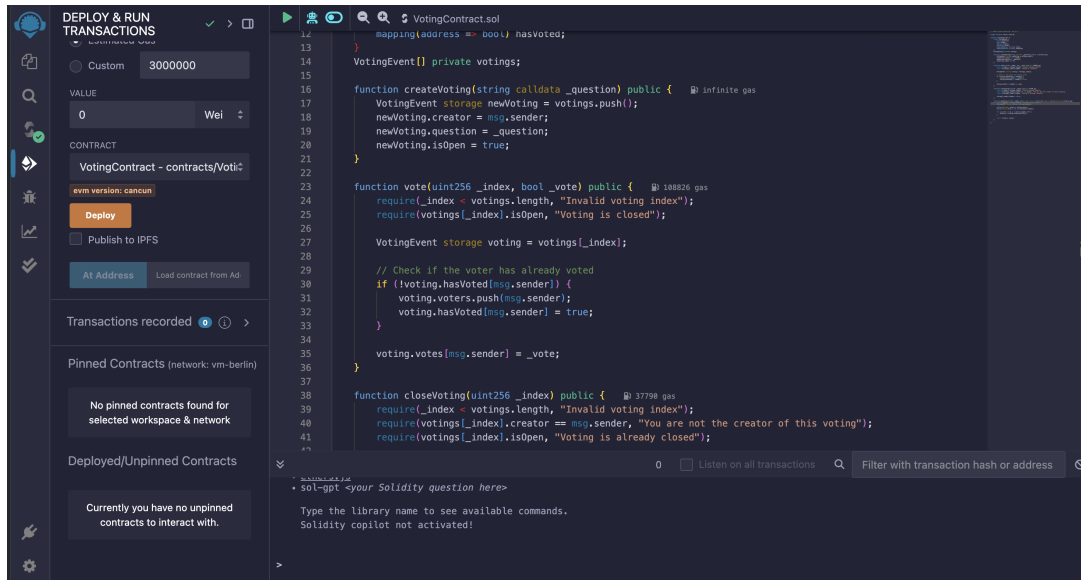
έναν με τις αντίστοιχες ψήφους τους. Αρχικά, ελέγχει αν ο δοσμένος δείκτης είναι έγκυρος, διασφαλίζοντας ότι αναφέρεται σε υπαρκτή ψηφοφορία. Στη συνέχεια, χρησιμοποιώντας τον πίνακα των ψηφοφόρων, δημιουργεί ένα νέο πίνακα votes όπου αποθηκεύει την ψήφο του κάθε ψηφοφόρου στον αντίστοιχο δείκτη. Τέλος, η συνάρτηση επιστρέφει τους δύο πίνακες.

```
1 function getVotes(uint256 _index) public view returns (address[] memory,  
   bool[] memory) {  
2     require(_index < votingEvents.length, "Invalid voting index");  
3     VotingEvent storage voting = votingEvents[_index];  
4  
5     address[] memory voters = voting.voters;  
6     bool[] memory votes = new bool[](voters.length);  
7  
8     for (uint256 i = 0; i < voters.length; i++) {  
9         votes[i] = voting.votes[voters[i]];  
10    }  
11  
12    return (voters, votes);  
13 }
```

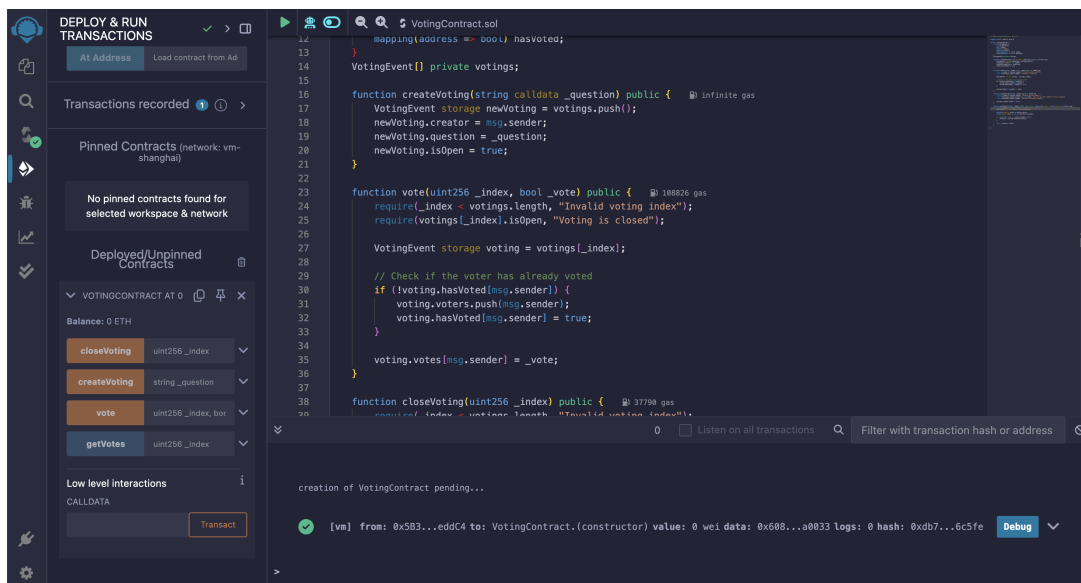
Εικόνα 6: Συνάρτηση προβολής ψήφων μιας ψηφοφορίας.

## 2 Στιγμιότυπα Οθόνης

Ακολουθούν σχετικά στιγμιότυπα οθόνης που δείχνουν την ορθή δημιουργία (deployment) των συμβολαίων στο Remix IDE.



Εικόνα 7: Screenshot του Remix IDE.



Εικόνα 8: Deployment του Smart Contract στο Remix VM περιβάλλον.

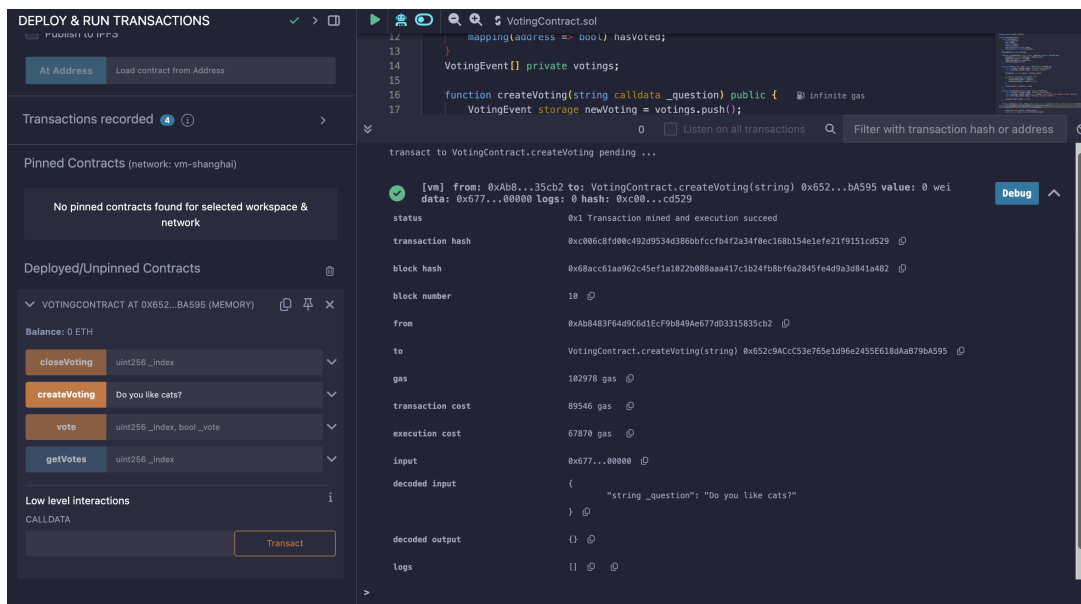
### 3 Παράδειγμα Εκτέλεσης

Το Remix παρέχει έναν εύκολο τρόπο στους χρήστες για να κάνουν deploy τα έξυπνα συμβόλαιά τους. Επίσης, δίνει τη δυνατότητα μέσω μιας φιλικής διεπαφής να καλούν τις συναρτήσεις τους.

#### 3.1 Δημιουργία Νέας Ψηφοφορίας

Θα δημιουργήσω μια ψηφοφορία (βλ. εικόνα 9) με την ερώτηση "Do you like cats?" και διεύθυνση λογαριασμού:

0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2



Εικόνα 9: Δημιουργία μιας ψηφοφορίας μέσω του Remix IDE.

#### 3.2 Δημιουργία Ψήφου σε Ψηφοφορία

Θα ψηφίσω true στην προηγούμενη ψηφοφορία η οποία είναι η μόνη που έχει δημιουργηθεί, άρα έχει δείκτη 0 στον πίνακα (βλ. εικόνα 10).

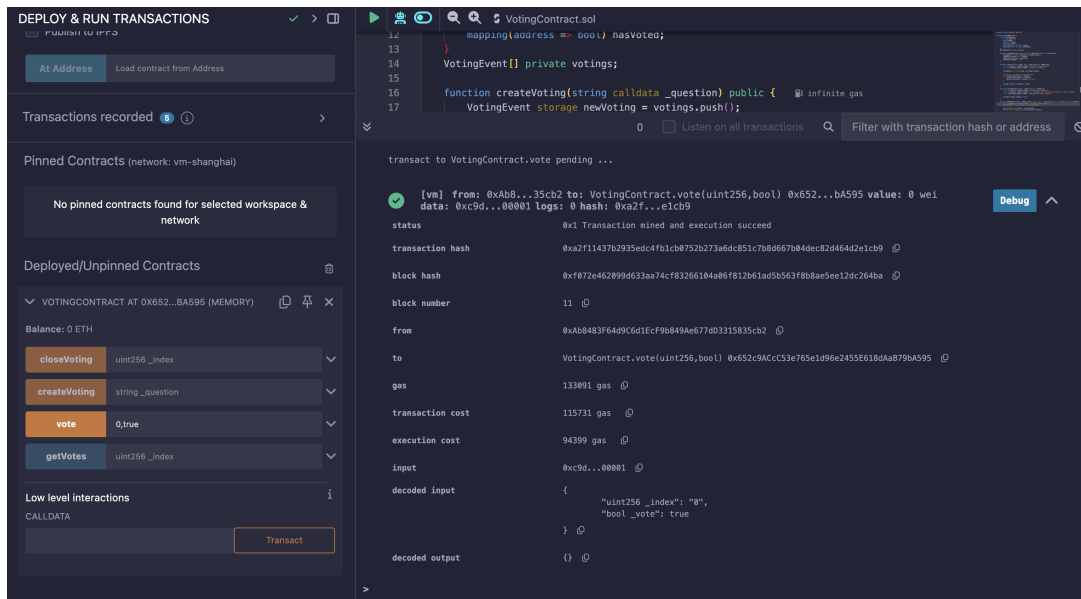
#### 3.3 Κλείσιμο Ψηφοφορίας

Αρχικά, θα προσπαθήσω να κλείσω την ψηφοφορία με δείκτη 0, από έναν διαφορετικό λογαριασμό (βλ. εικόνα 11). Παρατηρούμε πως δεν έχω το δικαίωμα και το ερώτημα απορρίπτεται. Στη συνέχεια, θα κλείσω την ψηφοφορία με τον λογαριασμό που είχε δημιουργηθεί (βλ. εικόνα 12).

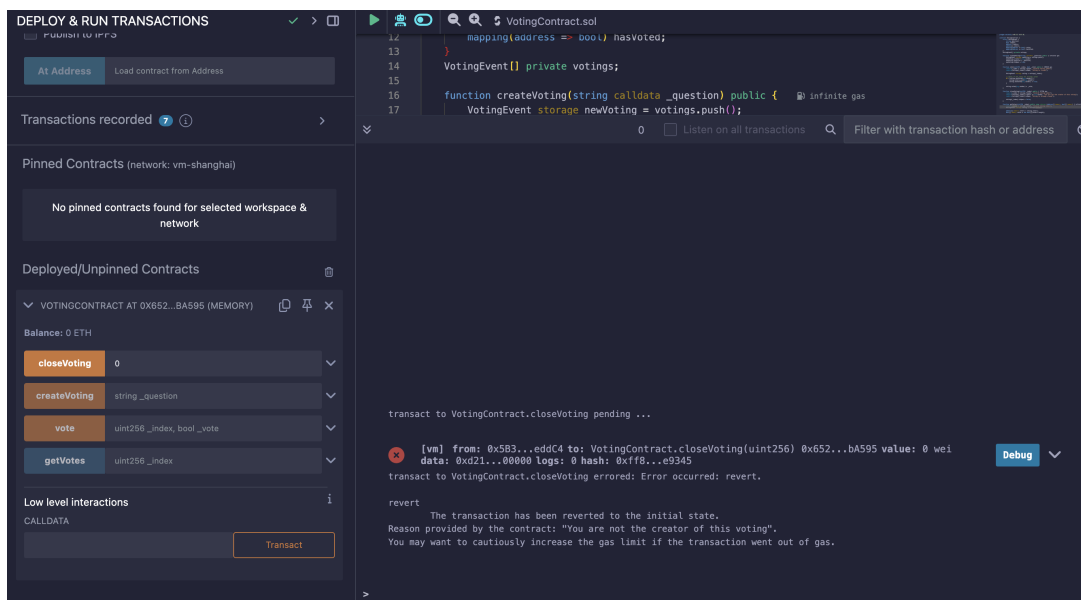
#### 3.4 Προβολή Αποτελεσμάτων

Για την προβολή των ψήφων μιας ψηφοφορίας αρκεί να κληθεί η συνάρτηση getVotes και να δοθεί ο δείκτης της ψηφοφορίας (βλ. εικόνα 13).

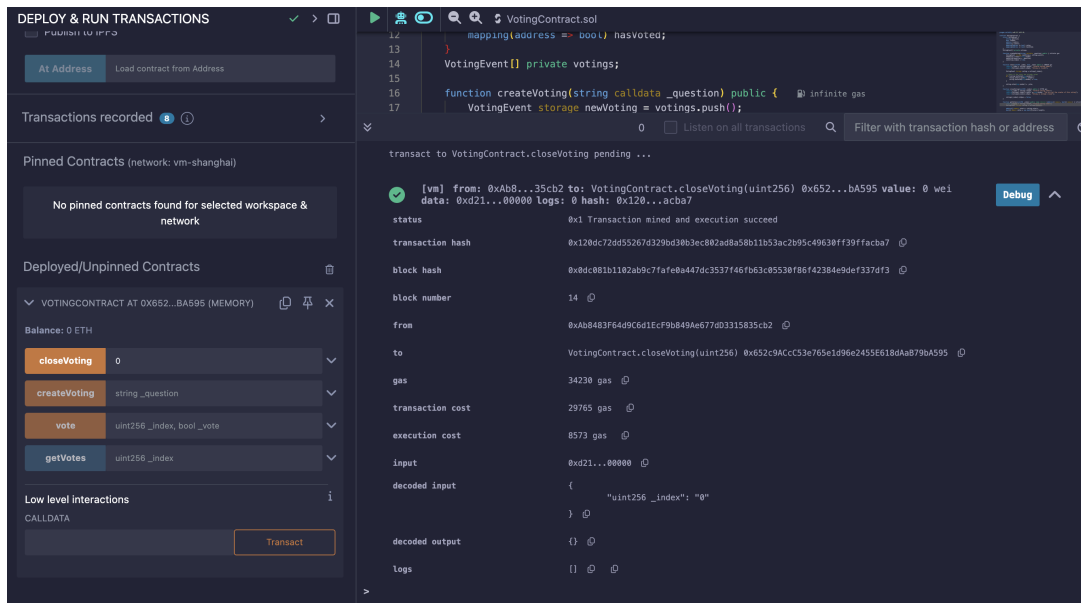




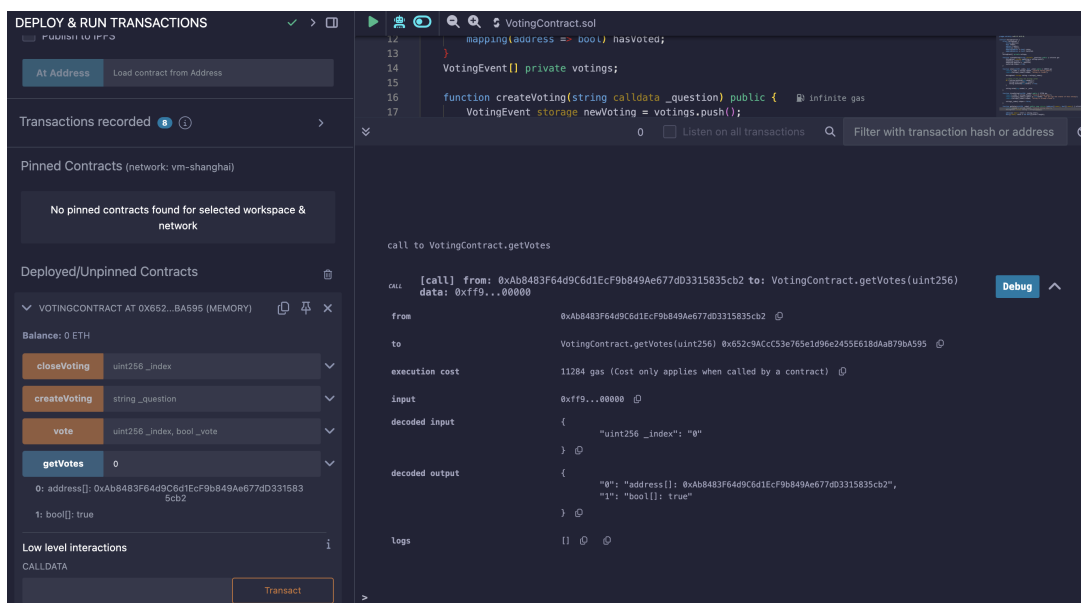
Εικόνα 10: Δημιουργία ψήφου μέσω του Remix IDE.



Εικόνα 11: Προσπάθεια κλεισίματος ψηφοφορίας από διαφορετικό λογαριασμό μέσω Remix IDE.



Εικόνα 12: Κλείσιμο ψηφοφορίας μέσω Remix IDE.



Εικόνα 13: Προβολή ψήφων μας ψηφοφορίας μέσω Remix IDE.