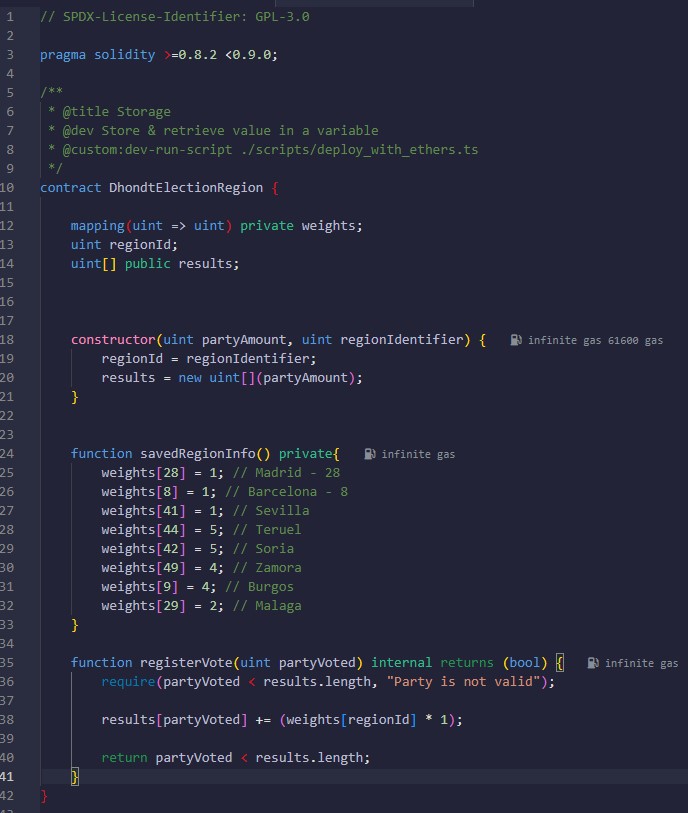
**Lab 4 - Inheritance, visibility and modifiers in Solidity**

*Group 22 – Arda Harman*

*Introduction*

We are having an election. In this election, we will be interacting with the following elements:

* Region
  + Voting can be done in different regions like Madrid, Terruel etc.
  + Each region will be represented with a code (details are at DHondtElectionRegion contract)
  + All of the regions will be represented with DHondtElectionRegion contract
* Polling Station
  + It represents the places where people are voting
  + For this exercise, we assume that we will be only having 1 polling station per region
  + Each polling station will be an instance of DhondtPollingStation contract
* President
  + It represents the the person who can control a polling station
  + It can open and close voting
  + It can also get the amount of votes FOR THAT SPECIFIC REGION
* Election
  + It represents the whole election process
  + It is an instance of the Election contract
  + It has a function which can create polling station (createPollingStation)
  + It has the data for the “total amount of the votes”
* Authority
  + It is the person who can control the “election instance”
  + It can Access “total amount of the votes”
  1. **- Contract DhondtElectionRegion**



regionId => Represents the id of the region of the DhondtElectionRegion instance

results => Represents the total amount of the votes each party recieved for that specific instance

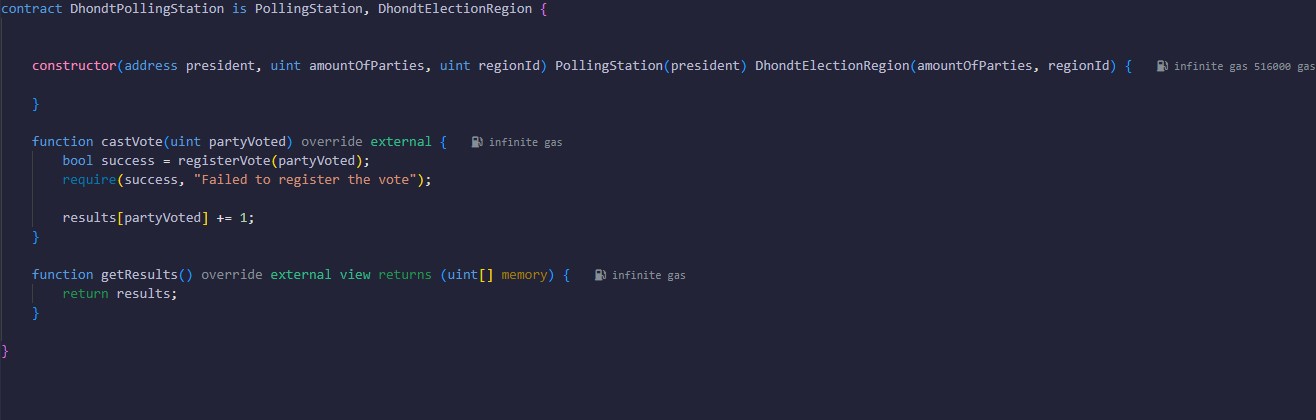
* 1. **- Contract PollingStation**



Here, I coded necesarry functions.

All functions are only executable by the president of that specific polling station

* 1. **- Contract DhondtPollingStation**

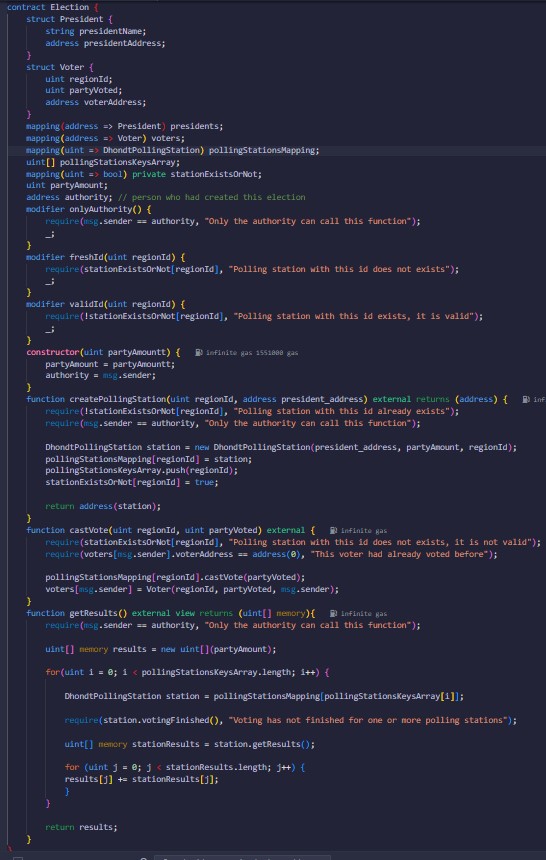


We cast the vote of the voter

But we cast, only if they haven’t voted before

We are able to check it by using the address of that voter

* 1. **- Contract Election**



Here, (since an election is the total all of those) we have maps for voters, presidents and stations

And we have every necessary function that we need to update our maps

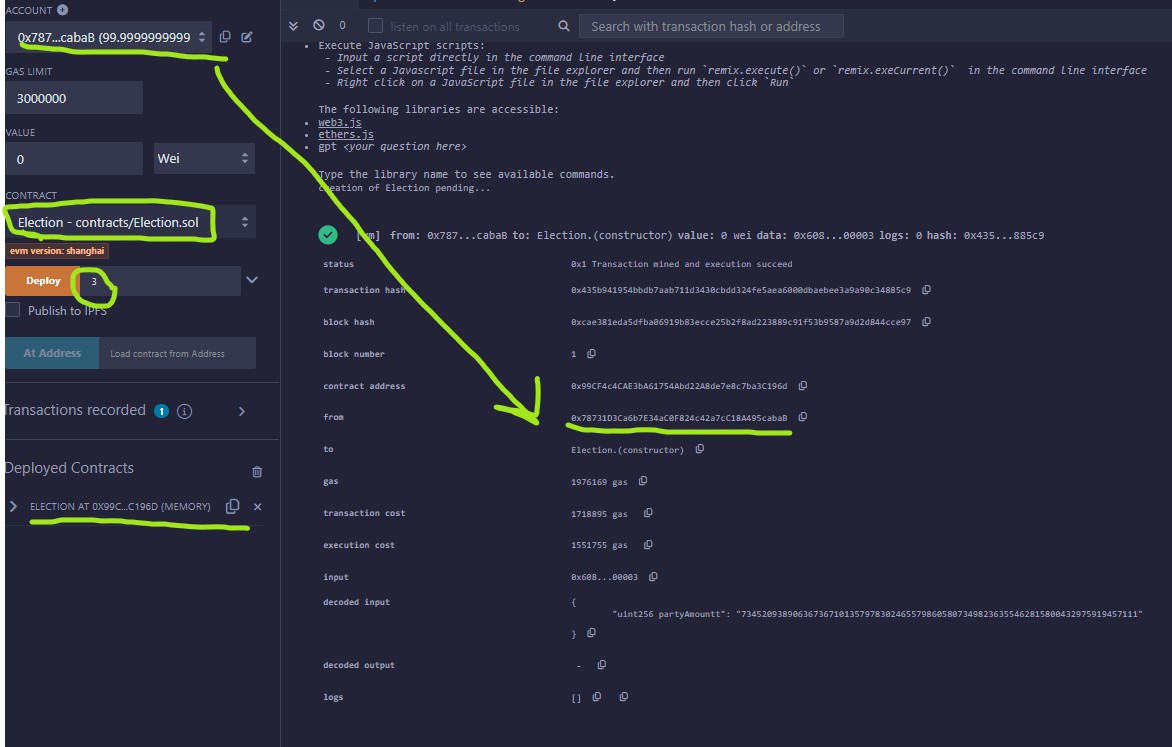
It is important to note that, all of these functions can be only executed by the authority

1. **- Contract Election**

Here are the addresses that I used for each element

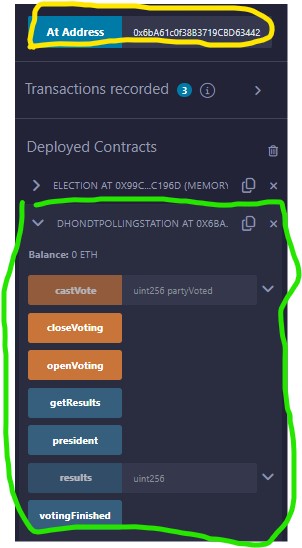
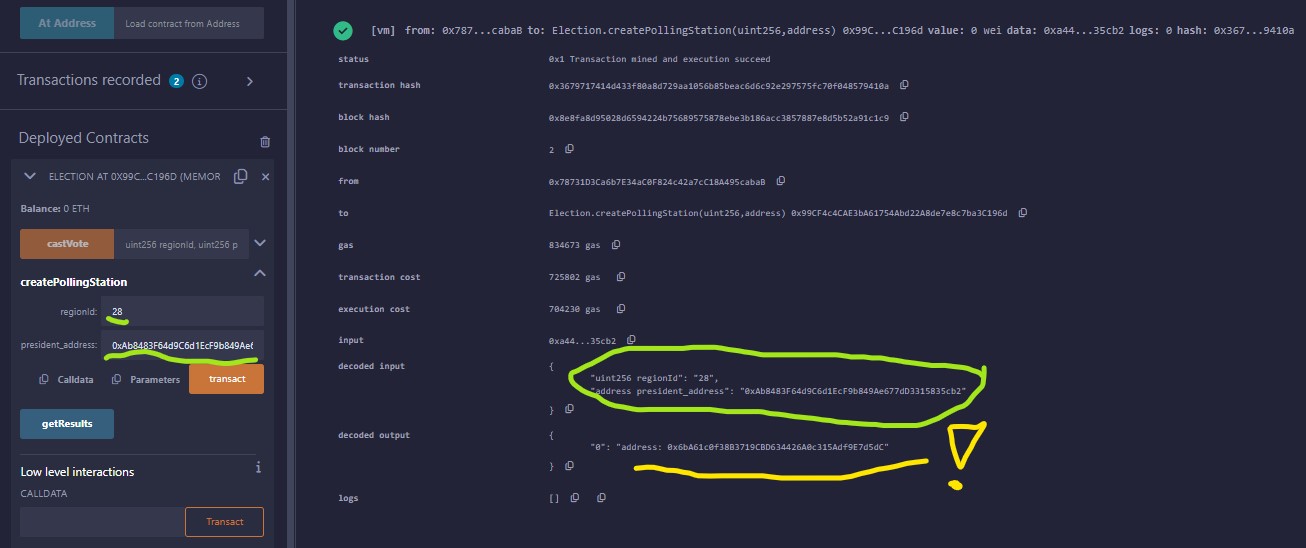
|  |  |
| --- | --- |
| **Role** | **Address** |
| Authority | 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB |
| President 1 | 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 |
| President 2 | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
| Voter 1 | 0x617F2E2fD72FD9D5503197092aC168c91465E7f2 |
| Voter 2 | 0x17F6AD8Ef982297579C203069C1DbfFE4348c372 |
| Voter 3 | 0x5c6B0f7Bf3E7ce046039Bd8FABdfD3f9F5021678 |
| Voter 4 | 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7 |
| Voter 5 | 0x1aE0EA34a72D944a8C7603FfB3eC30a6669E454C |

Deploying the election:

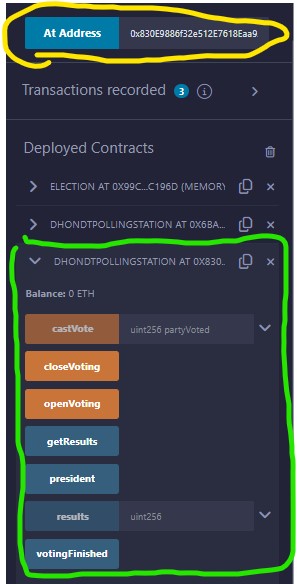
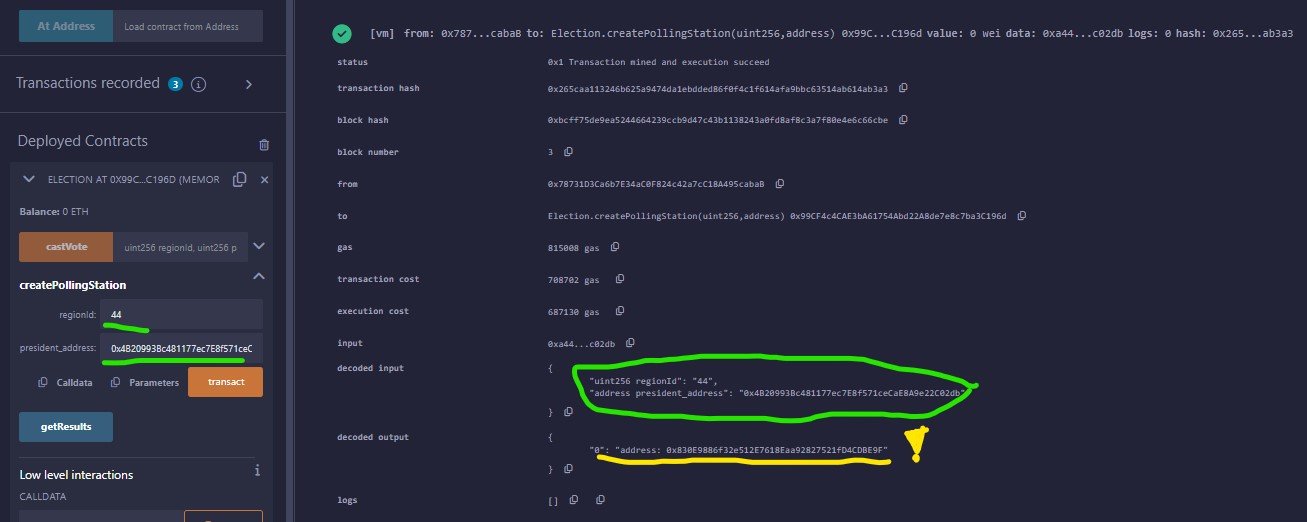


Creating polling station:

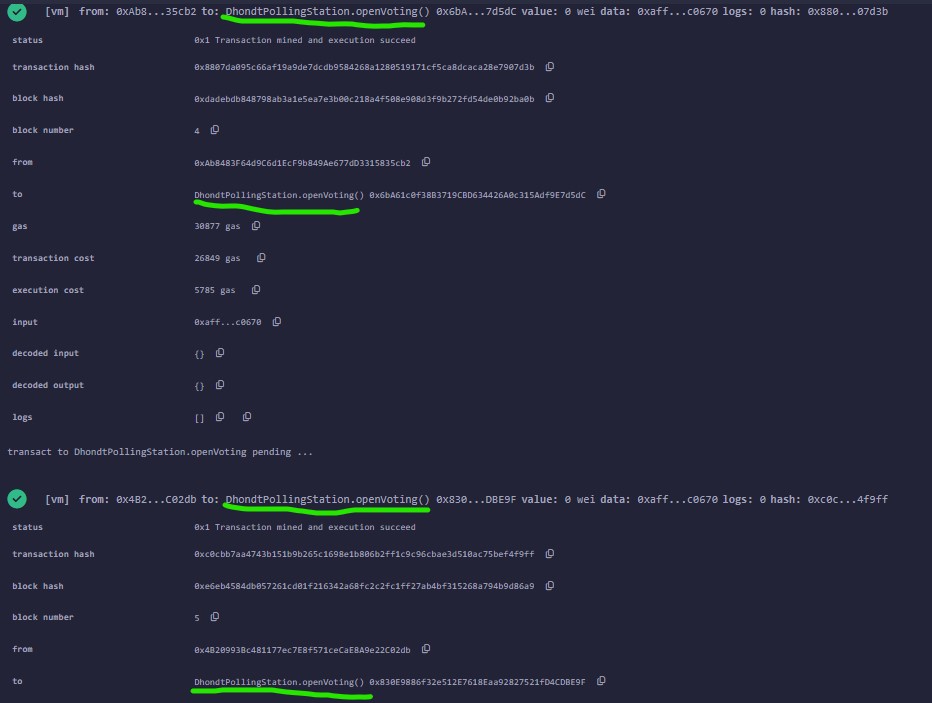
\* Madrid station



\* Terruel station

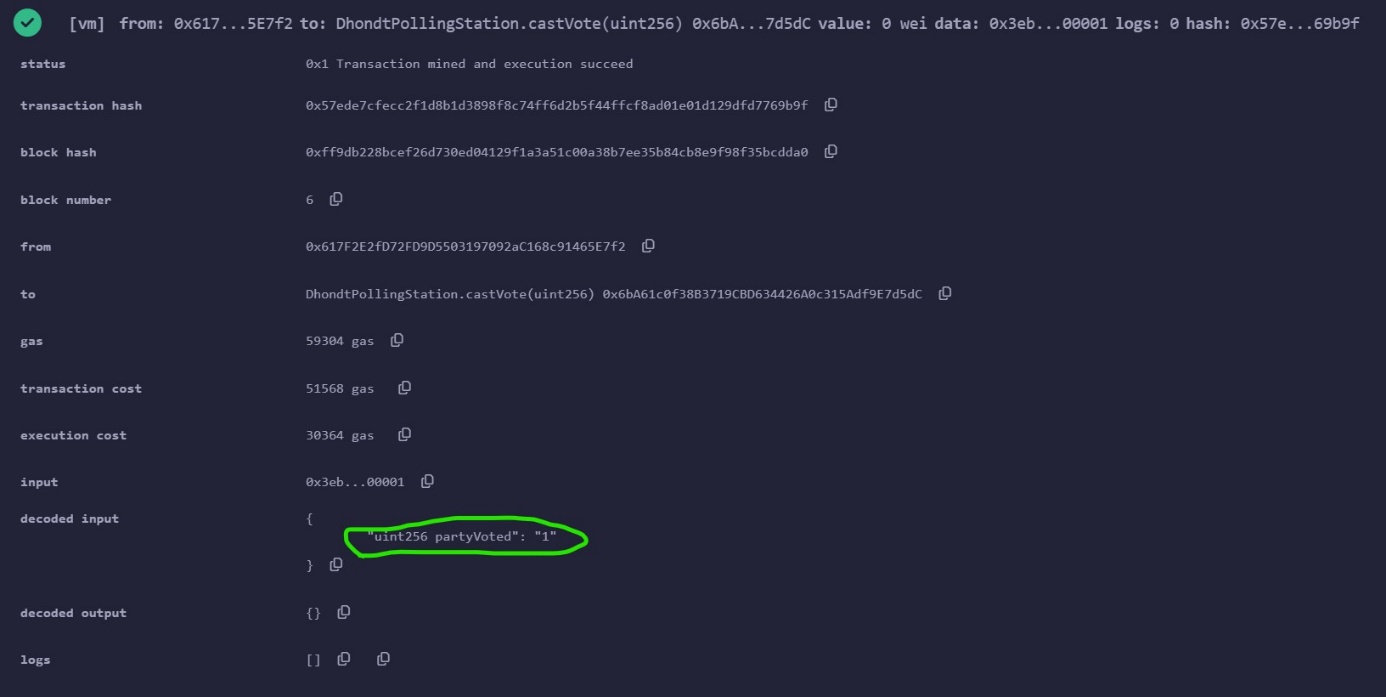


Invoking open voting

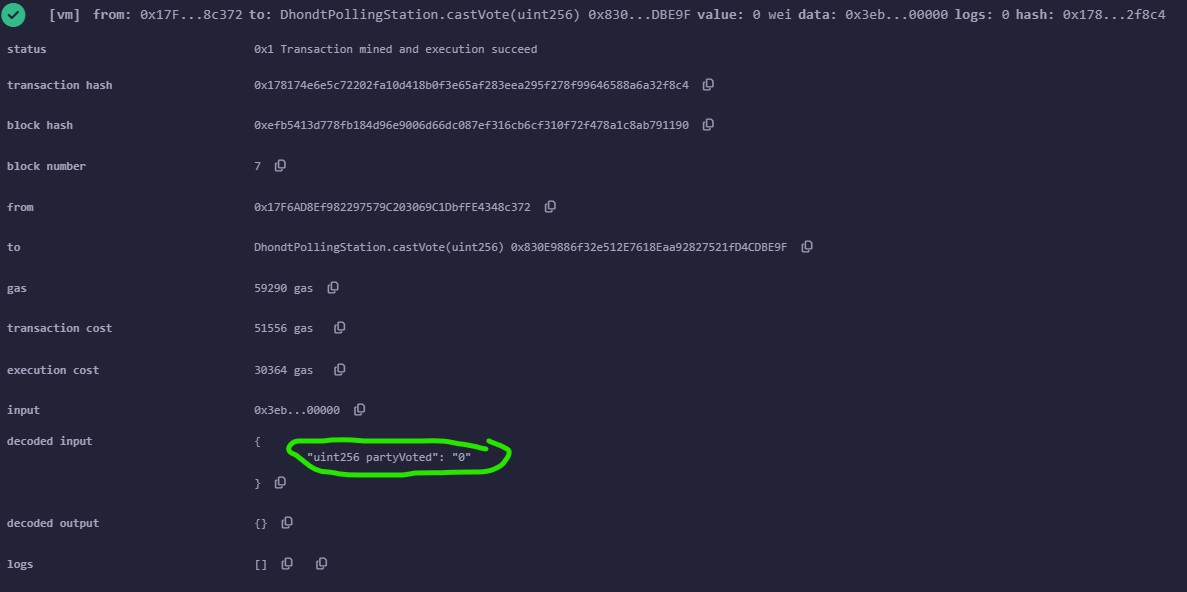


Casting votes:

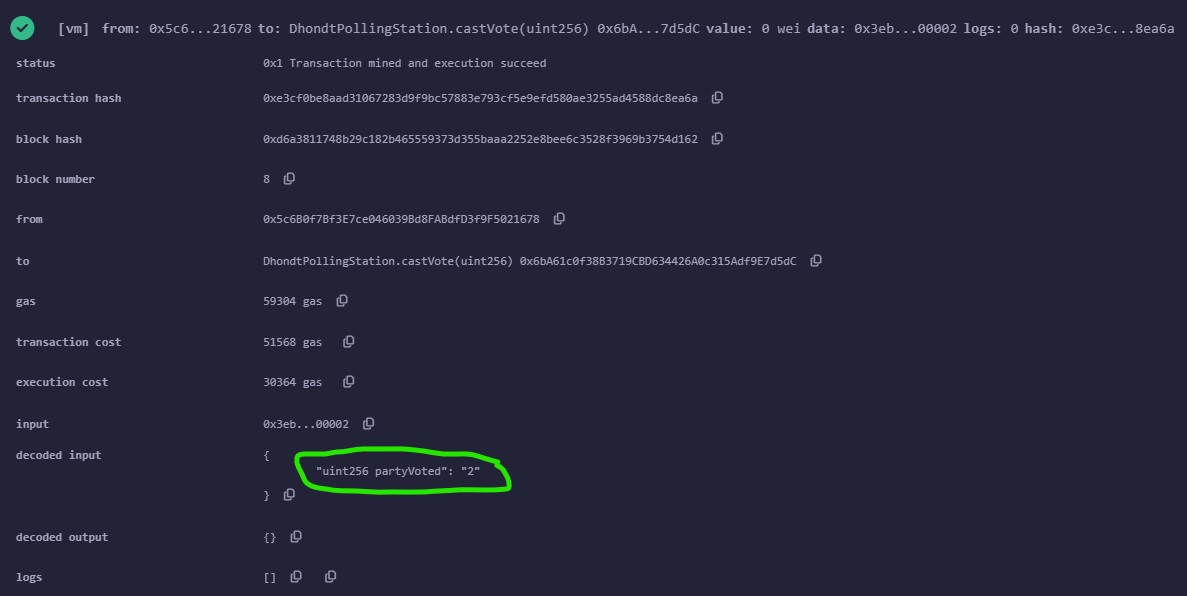
- Voter1: Party 1 in Madrid.



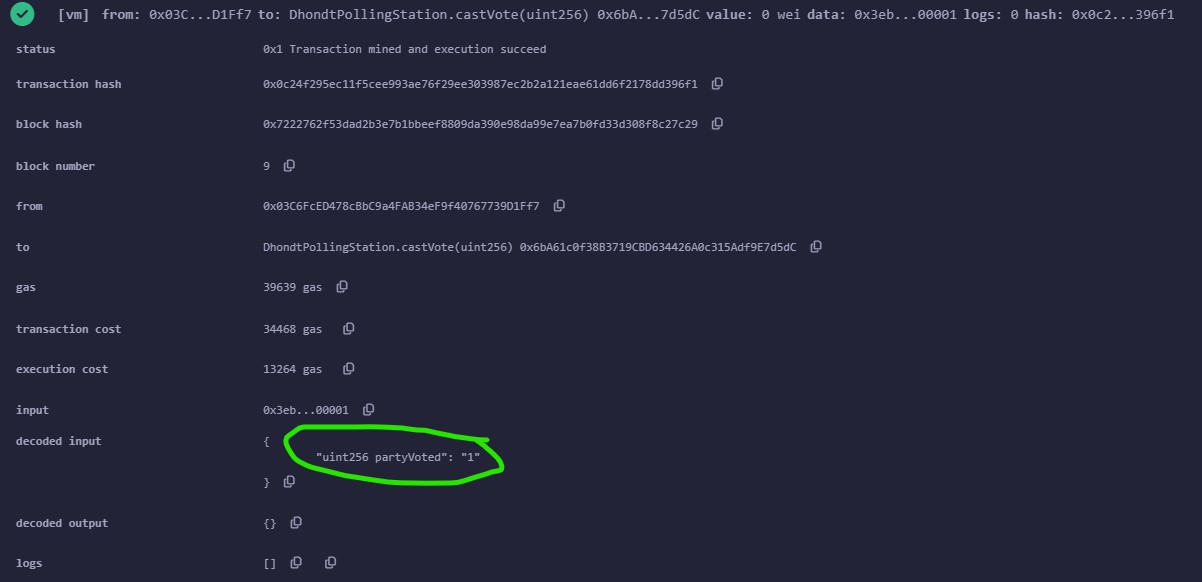
- Voter2: Party 0 in Teruel.



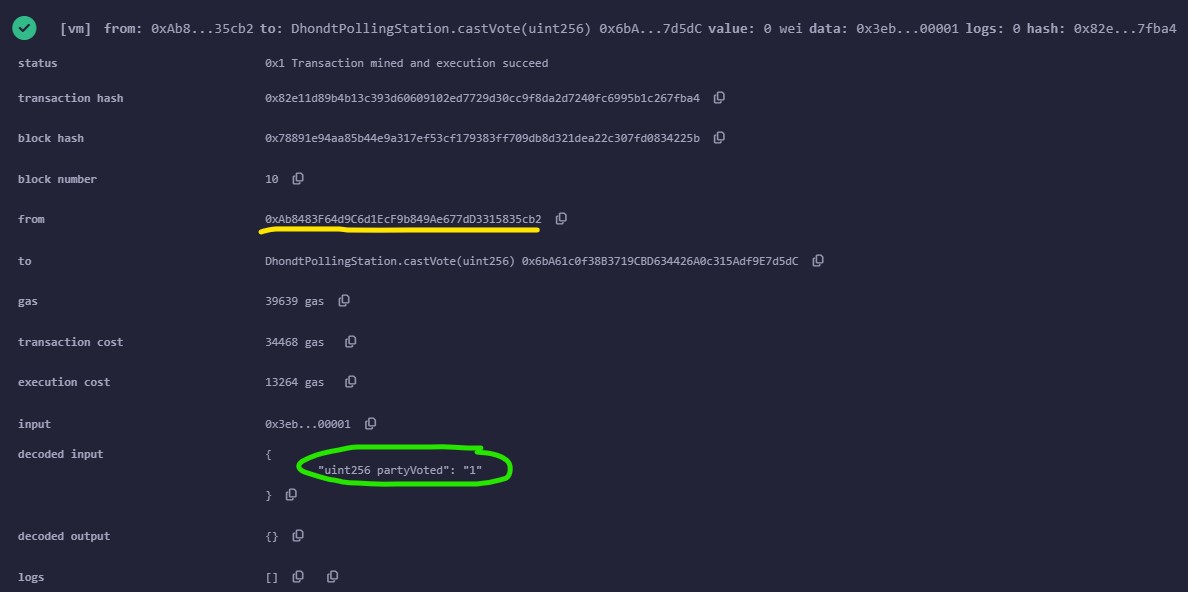
- Voter3: Party 2 in Madrid.



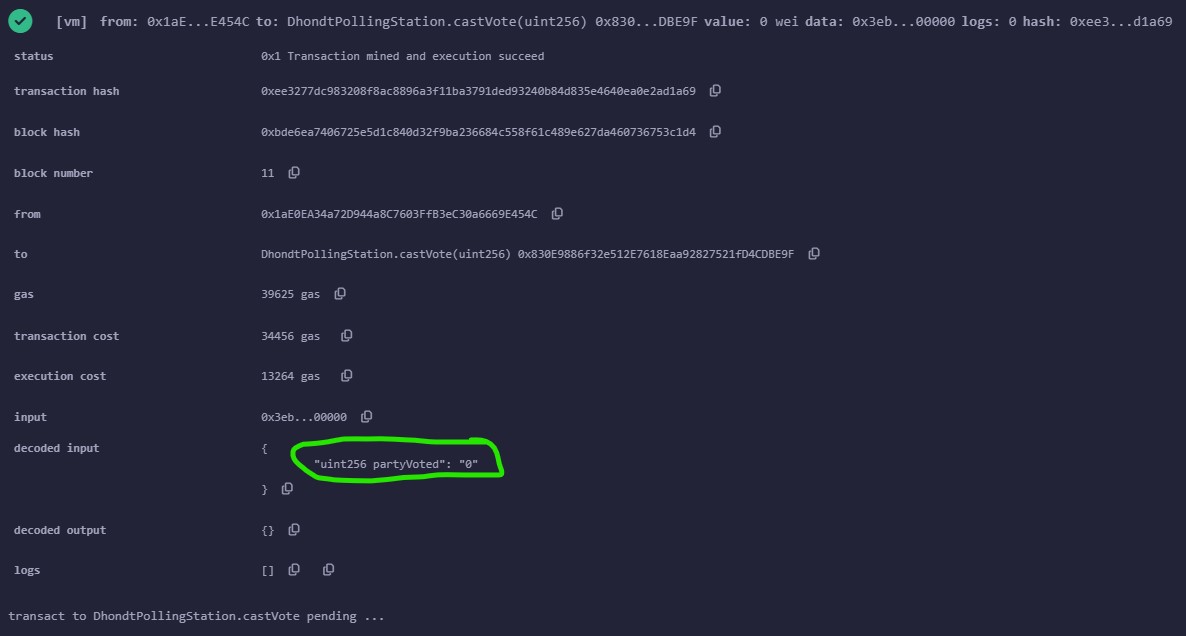
- Voter4: Party 1 in Madrid.

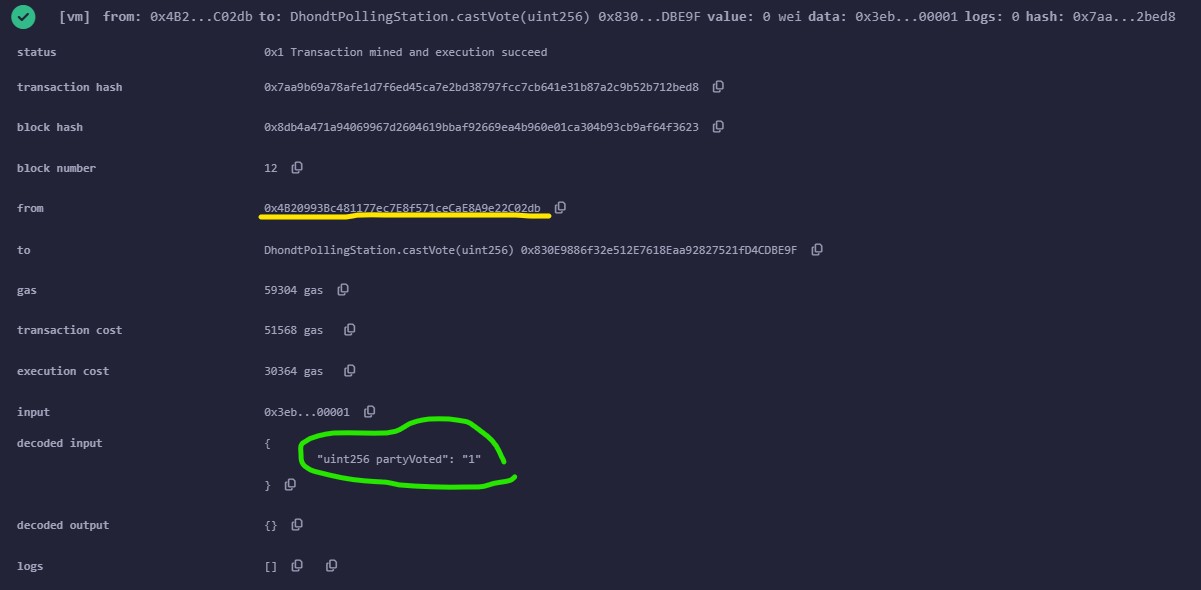


- President1: Party 1 in Madrid.

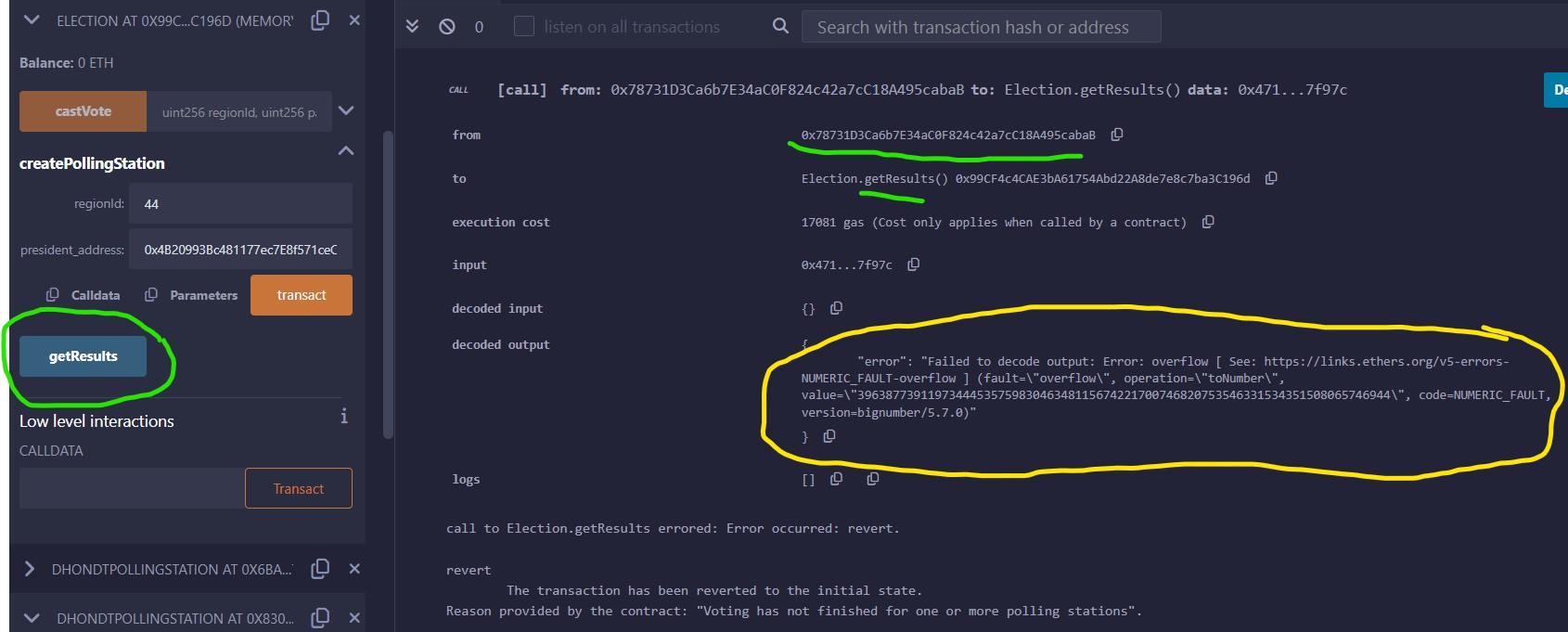


- Voter5: Party 0 in Teruel.

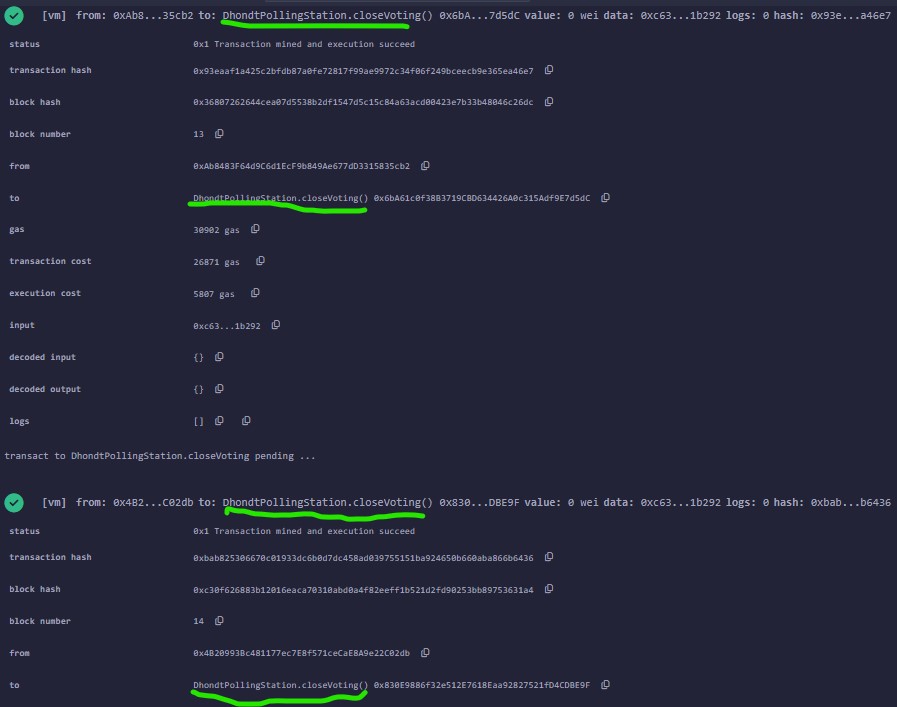


- President2: Party 1 in Teruel.

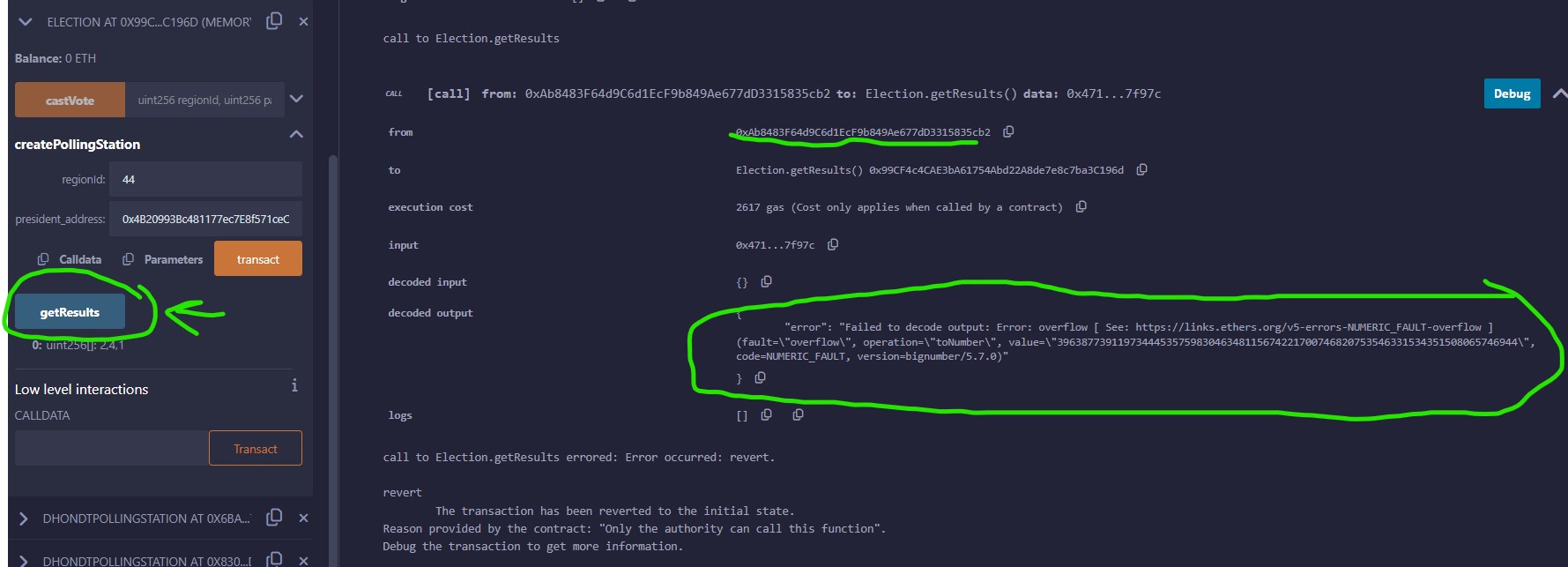
Invoking getResults with authority but without closing the elections:



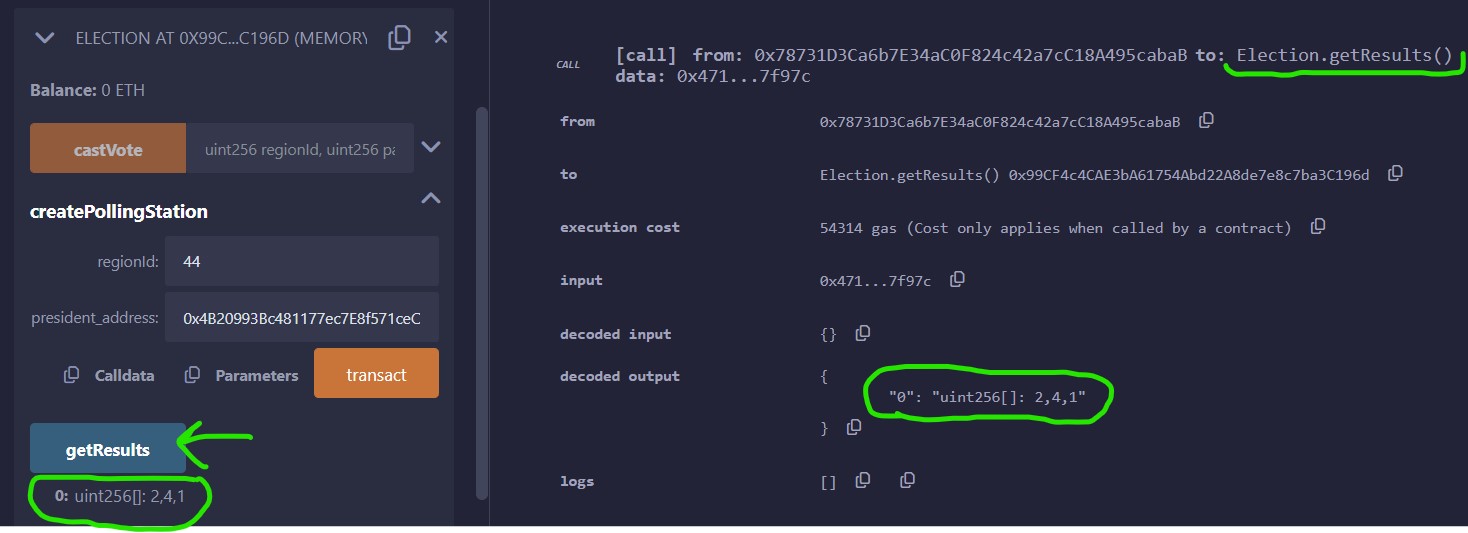
Closing the voting process:



Invoking getResult (after closing votings) with a president:



Invoking getResults (after closing votes) with the authority:



|  |  |
| --- | --- |
| **Party** | **Result** |
| Party 0 | 2 |
| Party 1 | 4 |
| Party 2 | 1 |