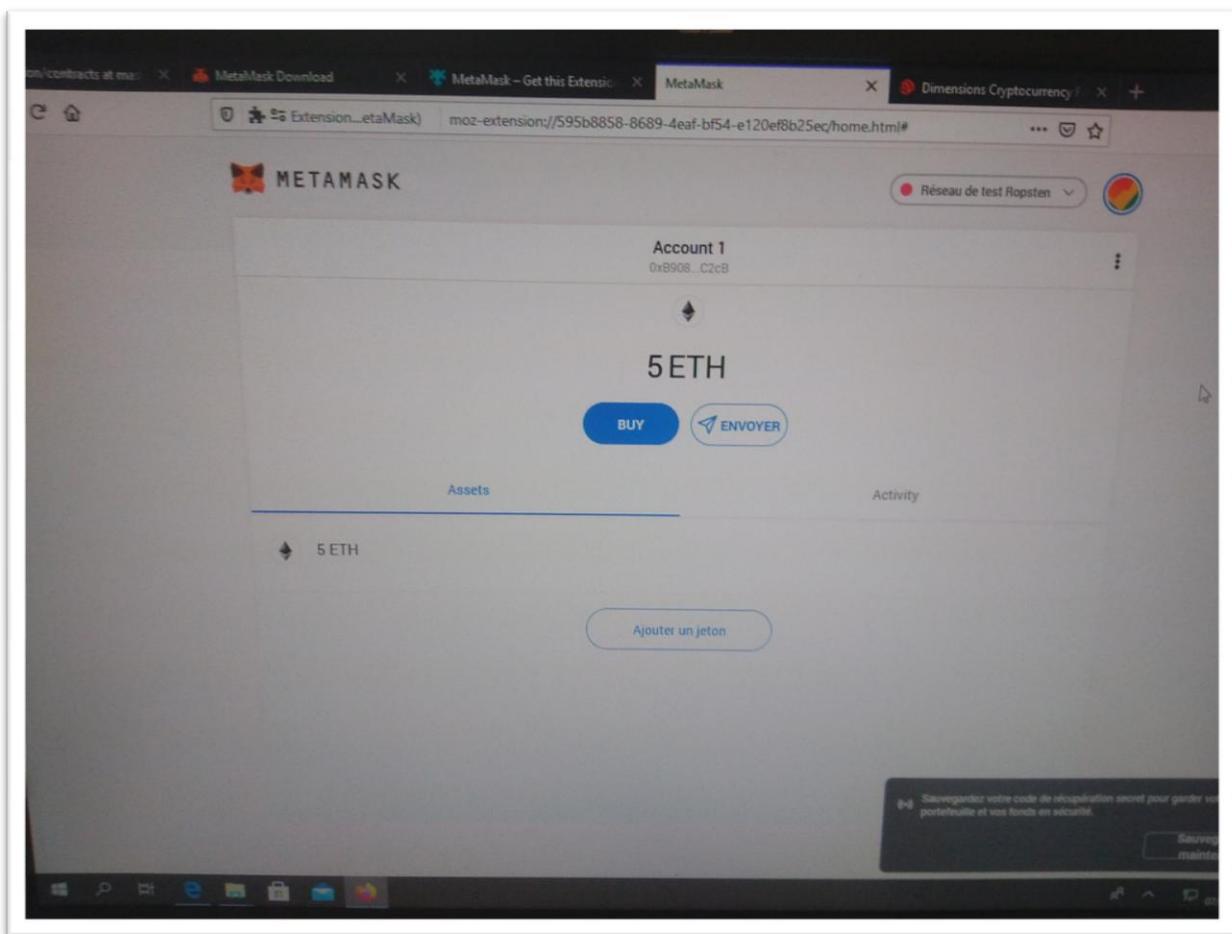


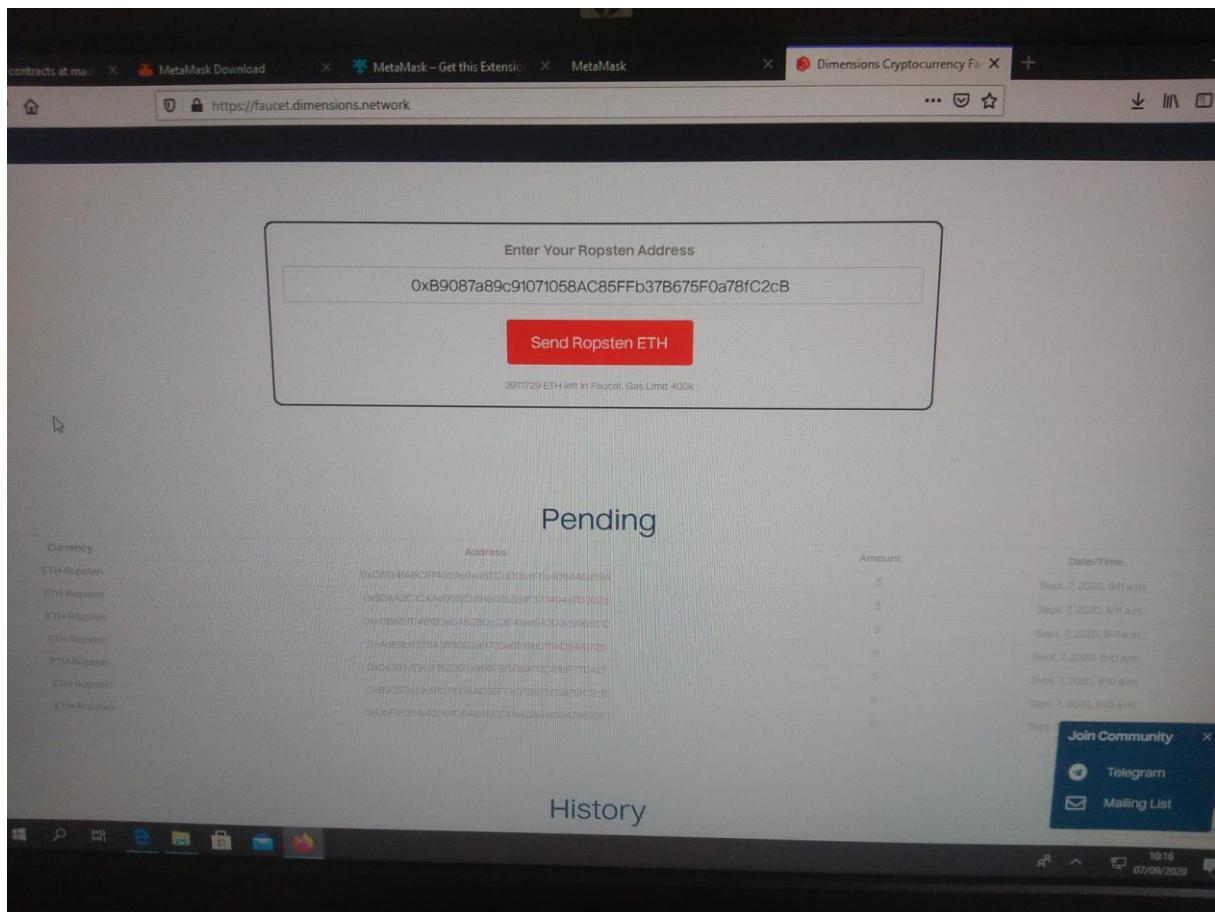
TRAVAUX PRATIQUE :

Développer, Déployer et Interagir avec un contrat intelligent sur Ethereum

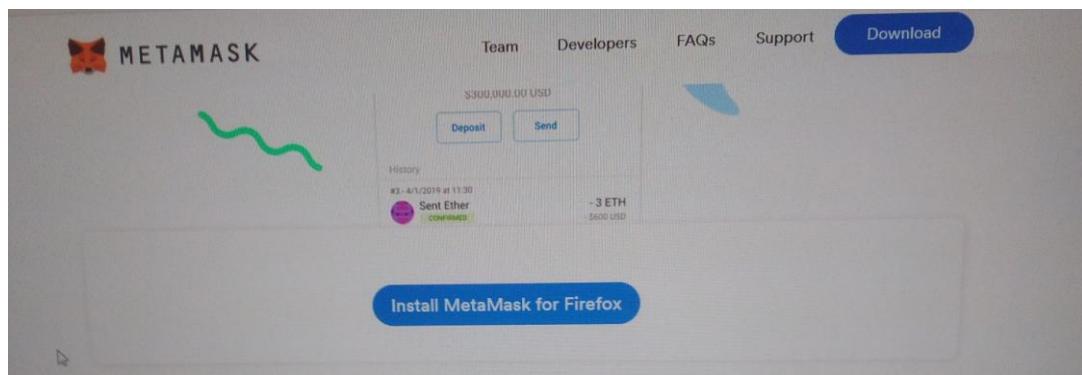
Première acquisition d'Ethers au travers de Réseau de test Ropsten



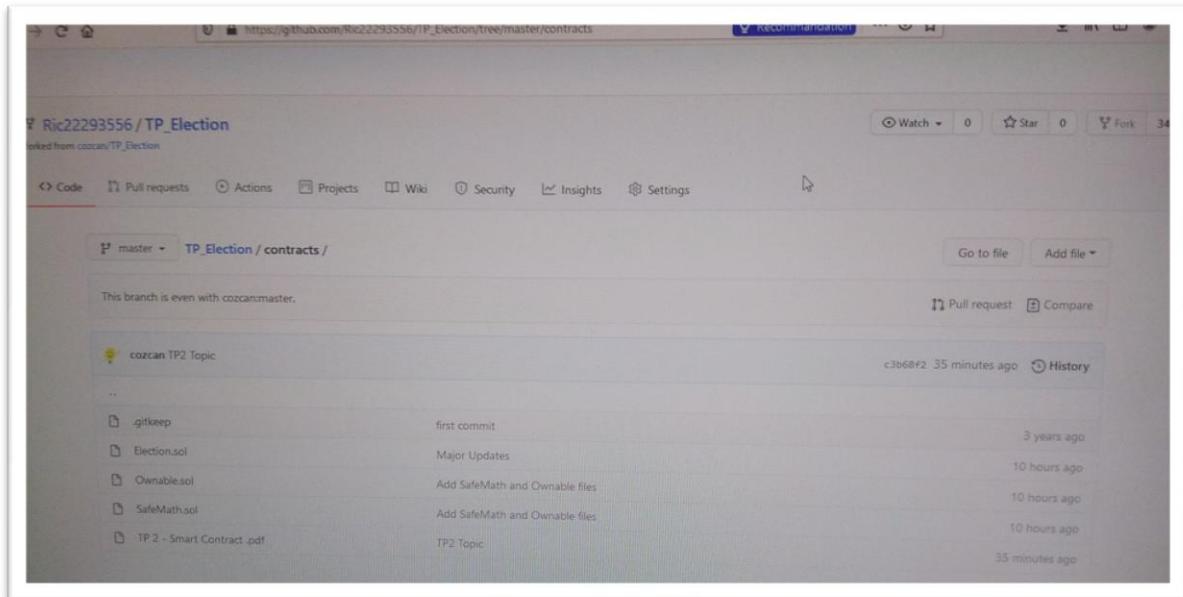
On peut alors observer le statut pending de la transaction, c'est-à-dire qu'elle est en attente d'exécution. Une fois la monnaie obtenue, celle-ci apparait dans le portefeuille.



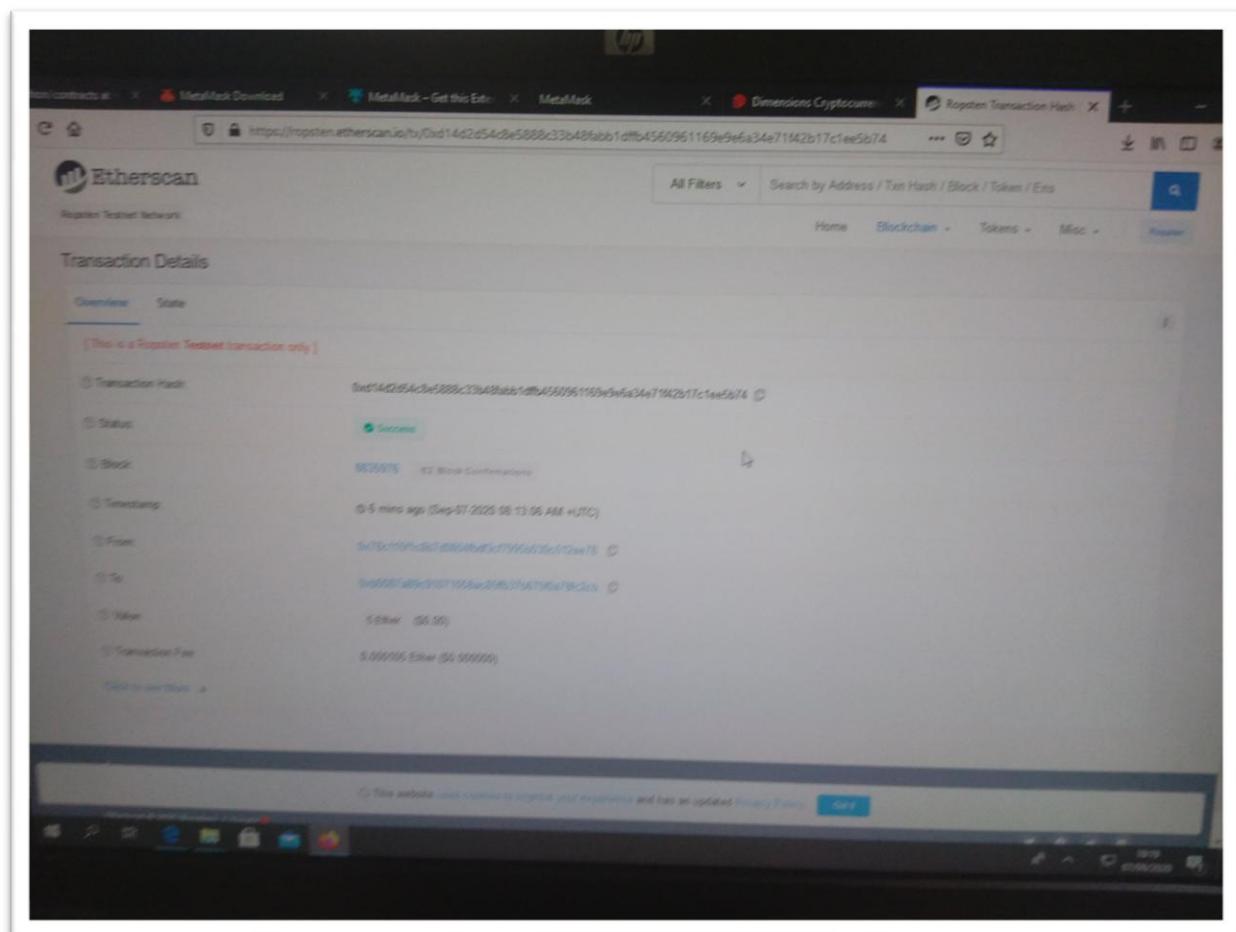
On installe alors Metamask pour firefox :



Une fois effectué, on peut alors charger les dossiers du tp dans notre compte personnel github :



On peut aussi voir le détail de la transaction à l'aide d'Etherscan :



Block #8635976

Overview

[This is a Ropsten Testnet block only]

- Block Height: 8635976
- Timestamp: 8 mins ago (Sep-07-2020 08:13:06 AM +UTC)
- Transactions: 16 transactions and 8 contract internal transactions in this block
- Mined by: 0xd34912efb0e7fedaedb9390990d7ef623e014fa in 18 secs
- Block Reward: 2.045189602 Ether (2 + 0.045189602)
- Uncles Reward: 0
- Difficulty: 539,487,739
- Total Difficulty: 31,436,483,676,960,759
- Size: 3,404 bytes
- Gas Used: 685,738 (8.61%)
- Gas Limit: 7,961,134
- Extra Data: poolin.com (Hex: 0x706f6fc696e2e6368d)

This website uses cookies to improve your experience, and has an updated Privacy Policy. [Get it](#)

On peut alors envoyer des Ethers à une autre personne grâce à sa clé publique :

Envoyer des ETH Annuler

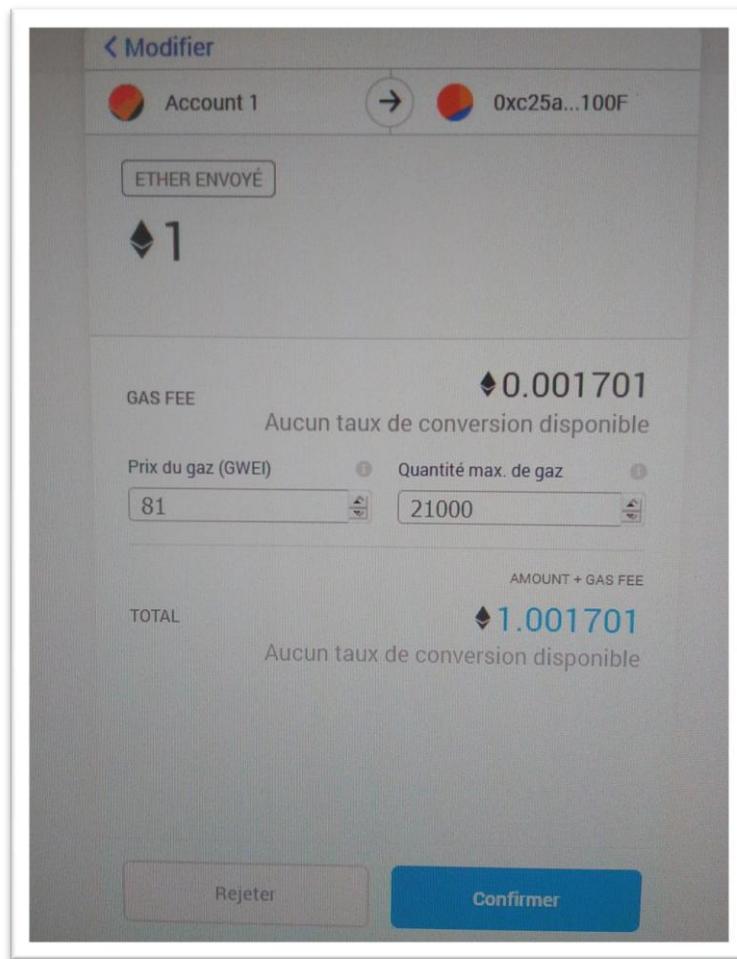
Nouvelle adresse détectée ! Cliquez ici pour ajouter à votre carnet d'adresses.

Actif:  ETH
Balance: 5 ETH

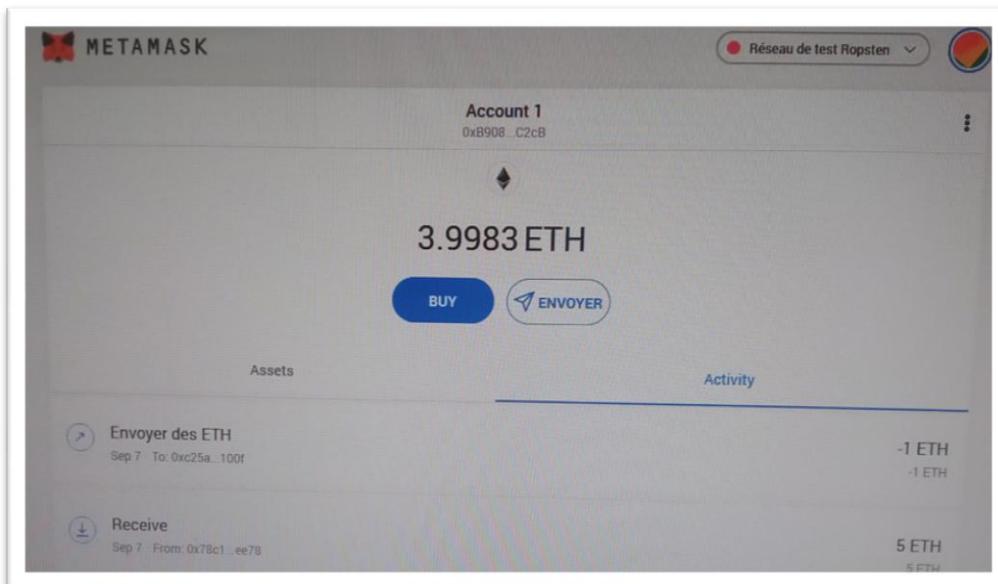
Montant: ETH
Aucun taux de conversion disponible

Frais de transaction: Prix du gaz (GWEI) Quantité max. de gaz

[Annuler](#) [Suivant](#)

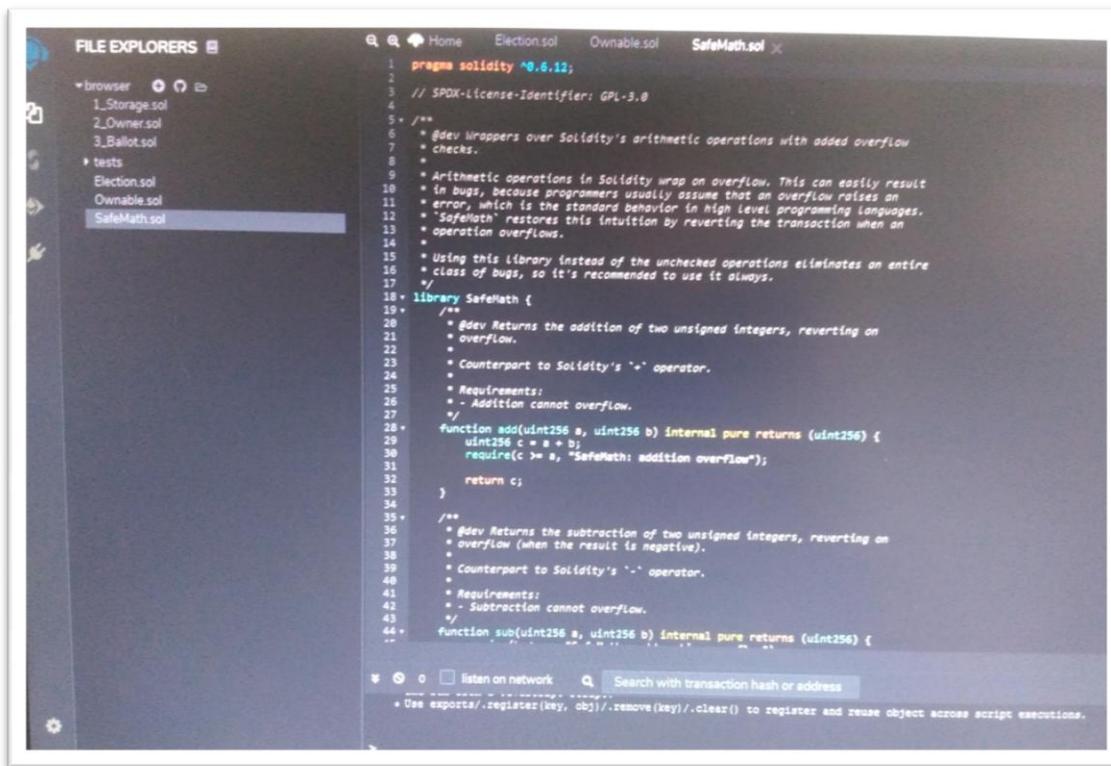


Après avoir envoyé les Ethers, forcément il y en a moins dans le portefeuille :



Les frais transactions diffèrent en fonction du nombre de transactions de users sur le réseau. Plus il y a de monde, plus il faut payer pour voir sa transaction s'effectuer dans les plus brefs délais. Cependant, il est difficile d'anticiper la congestion dans le réseau en mesurant les frais de gaz.

On peut alors se connecter sur Remix pour visualiser le code solidity d'un contrat :



```

FILE EXPLORERS ▾
browser ① ② ③
1_Storage.sol
2_Ownable.sol
3_Ballot.sol
tests
Election.sol
Ownable.sol
SafeMath.sol

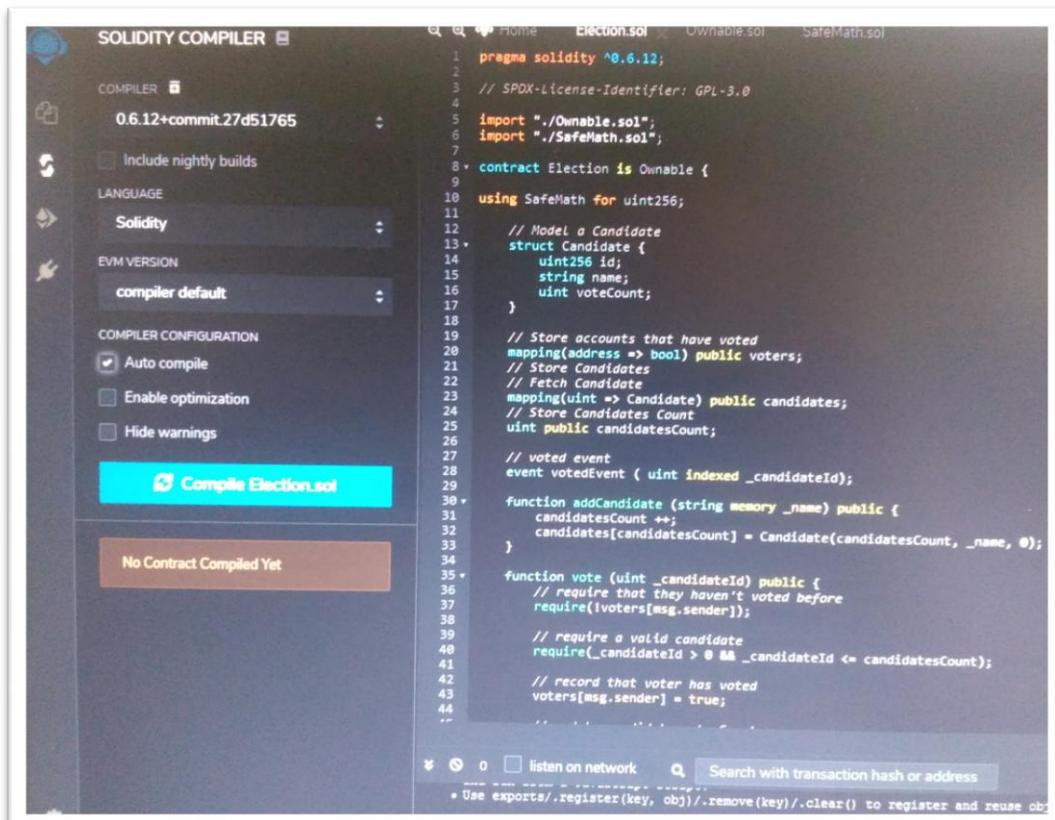
Home Election.sol Ownable.sol SafeMath.sol X

1 pragma solidity ^0.6.12;
2 // SPDX-License-Identifier: GPL-3.0
3 /**
4 * @dev Wrappers over Solidity's arithmetic operations with added overflow
5 * checks.
6 *
7 * Arithmetic operations in Solidity wrap on overflow. This can easily result
8 * in bugs, because programmers usually assume that an overflow raises an
9 * error, which is the standard behavior in high level programming languages.
10 * "SafeMath" restores this intuition by reverting the transaction when an
11 * operation overflows.
12 *
13 * Using this library instead of the unchecked operations eliminates an entire
14 * class of bugs, so it's recommended to use it always.
15 */
16 library SafeMath {
17     /**
18      * @dev Returns the addition of two unsigned integers, reverting on
19      * overflow.
20      *
21      * Counterpart to Solidity's `+` operator.
22      *
23      * Requirements:
24      *
25      * - Addition cannot overflow.
26     */
27     function add(uint256 a, uint256 b) internal pure returns (uint256) {
28         uint256 c = a + b;
29         require(c >= a, "SafeMath: addition overflow");
30         return c;
31     }
32     /**
33      * @dev Returns the subtraction of two unsigned integers, reverting on
34      * overflow (when the result is negative).
35      *
36      * Counterpart to Solidity's `-` operator.
37      *
38      * Requirements:
39      *
40      * - Subtraction cannot overflow.
41     */
42     function sub(uint256 a, uint256 b) internal pure returns (uint256) {
43
44 }

* ① ② ③ listen on network Search with transaction hash or address
* Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.

```

Une fois ajouté, le code doit être compilé :



```

SOLIDITY COMPILER ▾
COMPILER 0.6.12+commit.27d51765
Include nightly builds
LANGUAGE Solidity
EVM VERSION compiler default
COMPILE CONFIGURATION
Auto compile
Enable optimization
Hide warnings
Compile Election.sol

No Contract Compiled Yet

pragma solidity ^0.6.12;
// SPDX-License-Identifier: GPL-3.0
import "./Ownable.sol";
import "./SafeMath.sol";
contract Election is Ownable {
    using SafeMath for uint256;
    struct Candidate {
        uint256 id;
        string name;
        uint voteCount;
    }
    mapping(address => bool) public voters;
    struct Candidates {
        mapping(uint => Candidate) candidates;
        uint public candidatesCount;
    }
    event votedEvent(uint indexed _candidateId);
    function addCandidate(string memory _name) public {
        candidatesCount++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    function vote(uint _candidateId) public {
        // require that they haven't voted before
        require(!voters[msg.sender]);
        // require a valid candidate
        require(_candidateId > 0 && _candidateId <= candidatesCount);
        // record that voter has voted
        voters[msg.sender] = true;
    }
}

* ① ② ③ listen on network Search with transaction hash or address
* Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.

```

Enfin, il doit être distribué sur le réseau :

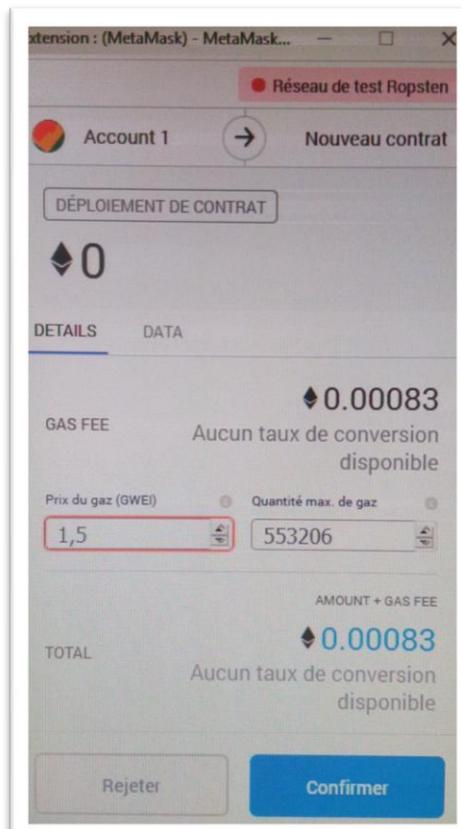
The screenshot shows the Truffle UI interface for deploying a Solidity contract. The environment is set to 'Injected Web3' with the network as 'Ropsten (3) network'. The account has a balance of 0xB90...fC2cB (3.998299 ether). The gas limit is set to 3000000. The value is 0 wei. The contract selected is 'Election - browser/Election.sol'. The code for the 'Election' contract is displayed:

```

pragma solidity ^0.6.12;
// SPDX-License-Identifier: GPL-3.0
import "./Ownable.sol";
import "./SafeMath.sol";
contract Election is Ownable {
    using SafeMath for uint256;
    struct Candidate {
        uint256 id;
        string name;
        uint voteCount;
    }
    mapping(address => bool) public voters;
    mapping(uint => Candidate) public candidates;
    uint public candidatesCount;
    event votedEvent(uint indexed _candidateId);
    function addCandidate(string memory _name) public {
        candidatesCount++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    function vote(uint _candidateId) public {
        require(!voters[msg.sender]);
        require(_candidateId > 0 && _candidateId <= candidatesCount);
        voters[msg.sender] = true;
    }
}

```

Below the code, a message says: "Currently you have no contract instances to interact with." There are options to "Publish to IPFS" or "At Address". The status shows "Transactions recorded" and "Deployed Contracts". A search bar at the bottom right is present.



On peut alors vérifier la bonne distribution du code :

Etherscan

Ropsten Testnet Network

All Filters ▾ Search by Address / Txn Hash / Block / Token / E

Home Blockchain ▾ Tokens ▾

Transaction Details

[Overview](#) [State](#)

[This is a Ropsten Testnet transaction only.]

⑦ Transaction Hash: [0x4ad4420bbffa6f44ae5ef5c2b19b24c8aee0886baddf8729172708c38a6333fb](#) ⓘ

⑦ Status: Success

⑦ Block: [8636202](#) 4 Block Confirmations

⑦ Timestamp: ⓘ 46 secs ago (Sep-07-2020 08:44:33 AM +UTC)

⑦ From: [0xb9087a89c91071058ac85ff37b675f0a78fc2b](#) ⓘ

⑦ To: [Contract [0x9cca507675d8a3e184766687889992a03527ab0c](#) Created] ⓘ ⓘ

⑦ Value: 0 Ether (\$0.00)

⑦ Transaction Fee: 0.000829809 Ether (\$0.000000)

[Click to see More](#) ↴

On retrouve notamment l'adresse du contrat dans le résumé de transaction, ici il s'agit de 0x9cca507675d8a3e184766687889992a03527ab0c :

[This is a Ropsten Testnet transaction only]

⑦ Transaction Hash: 0x4ad4420bbffa6f44ae5ef5c2b19b24c8aee0886baddf8729172708c38a6333fb

⑦ Status: Success

⑦ Block: 8636202 4 Block Confirmations

⑦ Timestamp: 46 secs ago (Sep-07-2020 08:44:33 AM +UTC)

⑦ From: 0xb9087a89c91071058ac85ffb37b675f0a78fc2cb

⑦ To: [Contract 0x9cca507675d8a3e184766687889992a03527ab0c Created] ✓

⑦ Value: 0 Ether (\$0.00)

⑦ Transaction Fee: 0.000829809 Ether (\$0.000000)

⑦ Gas Limit: 553,206

⑦ Gas Used by Transaction: 553,206 (100%)

⑦ Gas Price: 0.0000000015 Ether (1.5 Gwei)

ⓘ This website uses cookies to improve your experience and has an updated Privacy

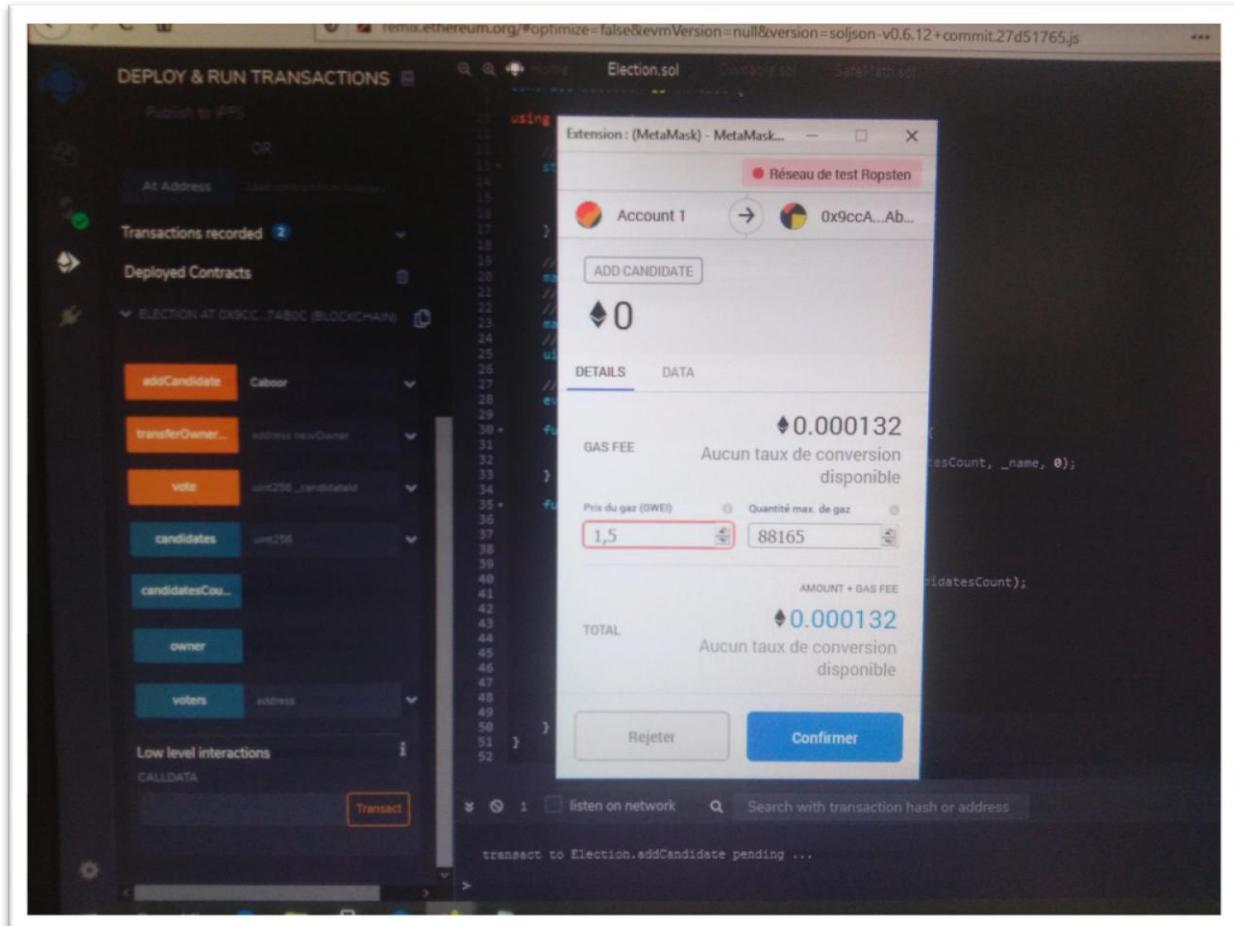
Une fois distribué, on peut charger le Bytes code et l'ABI dans github :

Ric22293556 Add files via upload 16f78dd 11 seconds ago 10 commits

build/contracts	first commit	3 years ago
contracts	TP2 Topic	1 hour ago
migrations	first commit	3 years ago
node_modules	first commit	3 years ago
src	first commit	3 years ago
test	first commit	3 years ago
ABI_RIC.txt	Add files via upload	29 seconds ago
Bytcode_RIC.txt	Add files via upload	11 seconds ago
bs-config.json	first commit	3 years ago
package-lock.json	first commit	3 years ago
package.json	first commit	3 years ago
truffle.js	first commit	3 years ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

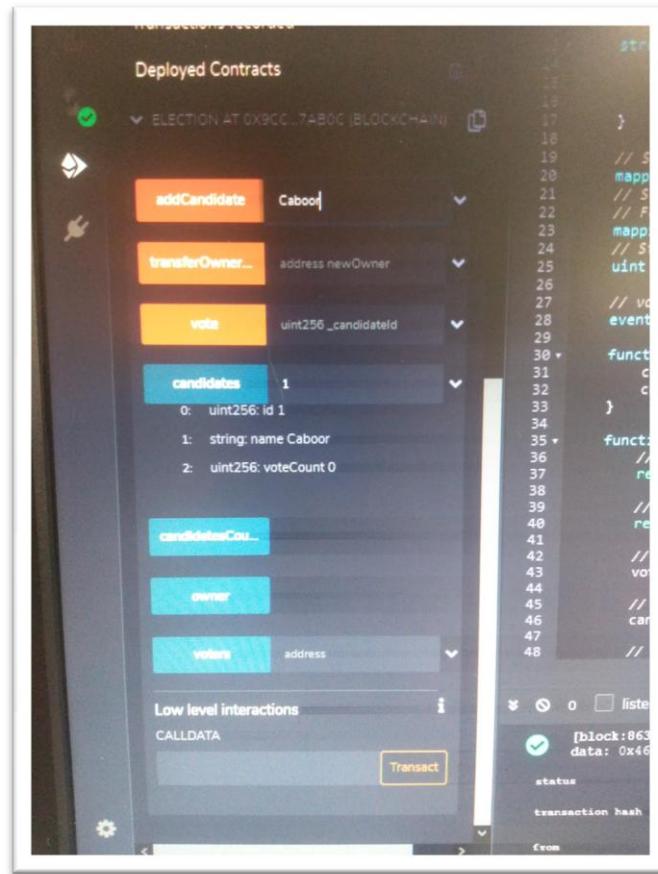
On cherche ensuite à ajouter un user au contrat nommé caboor :



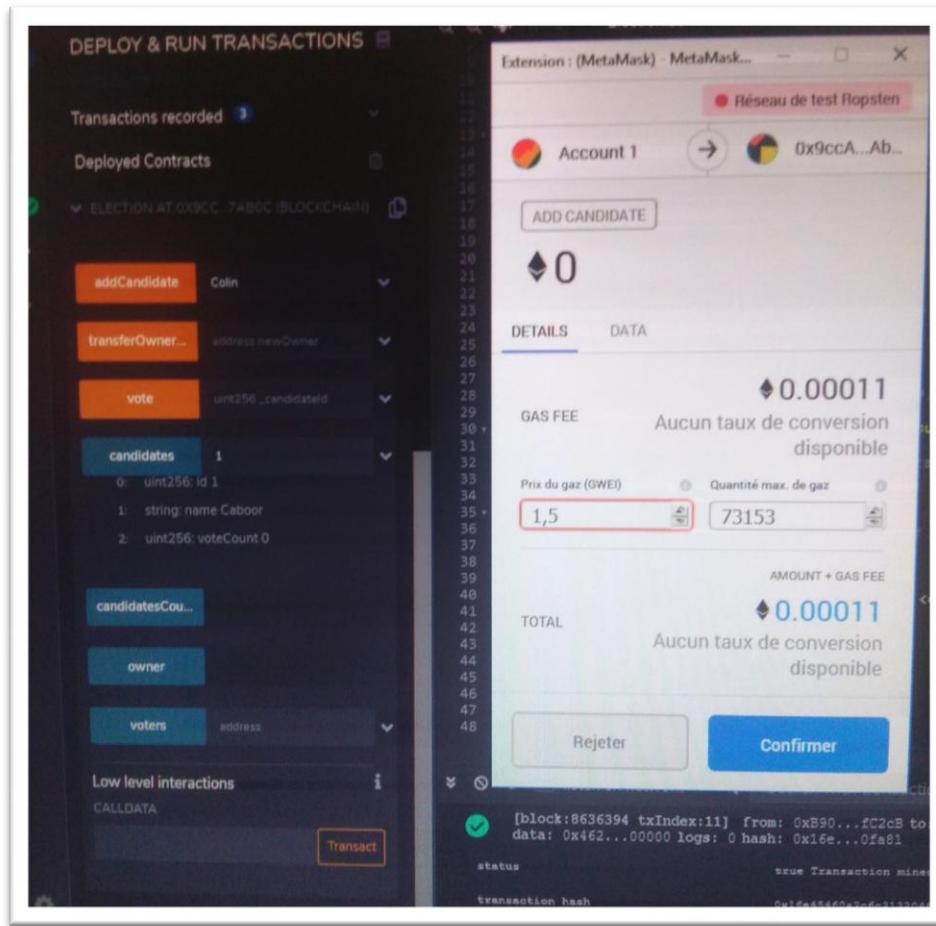
Le résultat de transaction donne :

Transaction Details	
Overview	State
[This is a Ropsten Testnet transaction only]	
⑦ Transaction Hash:	0x16e45460a3c6c3132046b38c5a3ccf0335200c8698f99d5fd0fad3f48280fa81 ⓘ
⑦ Status:	Success
⑦ Block:	8636394 3 Block Confirmations
⑦ Timestamp:	⌚ 36 secs ago (Sep-07-2020 09:05:13 AM +UTC)
⑦ From:	0xb9087a89c91071058ac85ffb37b675f0a78fc2cb ⓘ
⑦ To:	Contract 0x9cca507675d8a3e184766687889992a03527ab0c ⓘ
⑦ Value:	0 Ether (\$0.00)
⑦ Transaction Fee:	0.000130023 Ether (\$0.000000)
⑦ Gas Limit:	88,165
⑦ Gas Used by Transaction:	86,682 (98.32%)
⑦ Gas Price:	0.0000000015 Ether (1.5 Gwei)

On peut alors vérifier sur remix la bonne création du user. Son id est 1. On voit notamment qu'il ne possède pour le moment aucun vote :



On ajoute ensuite un deuxième user nommé colin, qui aura alors un id 2:



Son résultat de transaction donne :

Overview	State
<small>[This is a Ropsten Testnet transaction only]</small>	
⑦ Transaction Hash:	0x8d00a6a7101b6df2821fa48bce2de0897f511594434adad4cf5624e65241c01a 🔗
⑦ Status:	Success
⑦ Block:	8636545 1 Block Confirmation
⑦ Timestamp:	⌚ 32 secs ago (Sep-07-2020 09:24:38 AM +UTC)
⑦ From:	0xb9087a89c91071058ac85ffb37b675f0a78fc2cb 🔗
⑦ To:	Contract 0x9cca507675d8a3e184766687889992a03527ab0c ✓ 🔗
⑦ Value:	0 Ether (\$0.00)
⑦ Transaction Fee:	0.000107505 Ether (\$0.000000)
Click to see More ↓	

The screenshot shows the Truffle UI interface. At the top, it displays "Transactions recorded 3". Below this is a section titled "Deployed Contracts" with a dropdown menu showing "ELECTION AT 0x3CC7AB0C (BLOCKCHAIN)". The main area contains several contract methods listed as buttons:

- addCandidate**: Colin
- transferOwner**: address newOwner
- vote**: uint256 _candidateId
- candidates**: 2
 - 0: uint256 id 2
 - 1: string: name Colin
 - 2: uint256 voteCount 0
- candidateCount**
- owner**
- votes**: address

Below these buttons, there is a section titled "Low level Interactions" with a "CALLDATA" button and a "Transact" button.

On the right side of the screen, there is a vertical stack of code snippets, likely generated by the compiler, starting with "str" and ending with "transaction hash".

On peut alors récupérer l'adresse du propriétaire du contrat qui ici est 0xB9087a89c91071058AC85FFb37B675F0a78fC2cB :

The screenshot shows the Etherscan interface for the Ropsten Testnet Network. At the top, there's a search bar and navigation links for Home, Blockchain, Tokens, Misc, and Report. The address 0xB9087a89c91071058AC85FFb37B675F0a78fC2cB is displayed with a balance of 3.997231663 Ether. Below the address, there are sections for Overview (Balance: 3.997231663 Ether) and More Info (My Name Tag: Not Available). The Transactions section shows the latest 5 from a total of 5 transactions, with details like Txn Hash, Block, Age, From, To, Value, and Gas Fee.

On peut aussi ajouter un vote à un utilisateur :

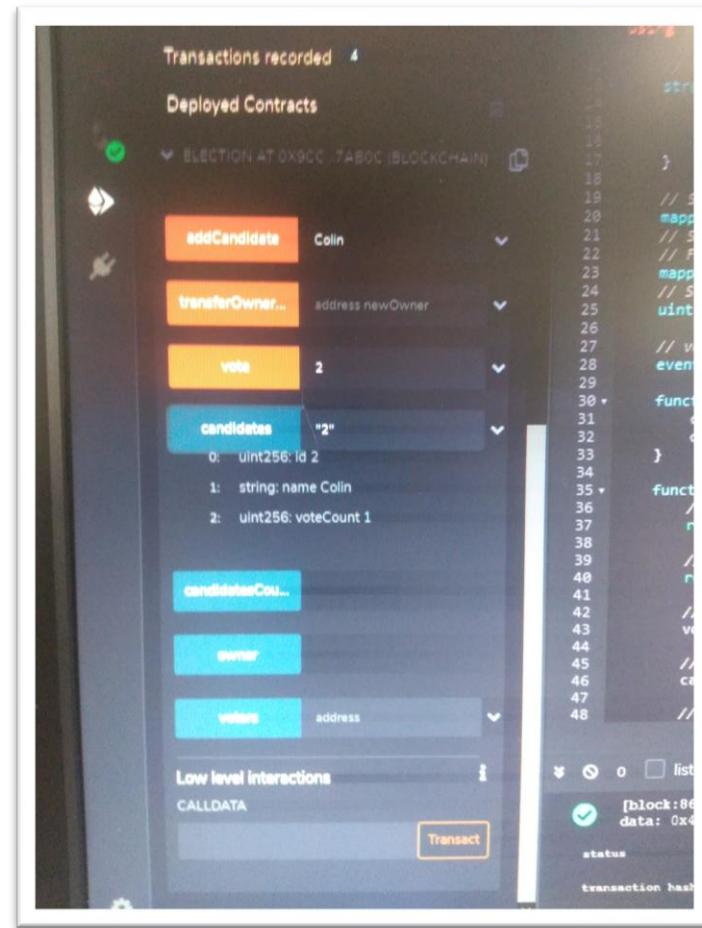
The screenshot shows the MetaMask extension interface for the Ropsten test network. It displays a transaction record for 'ELECTION AT 0x9CC...7AB0C (BLOCKCHAIN)'. The transaction details show a 'vote' action for candidate 'Colin' with value '2'. The transaction status is shown as 'Confirmed' with a green checkmark. The transaction hash is 0x16e45460a3c6c3132. The transaction fee is 0.000099 Ether.

Résultat de transaction :

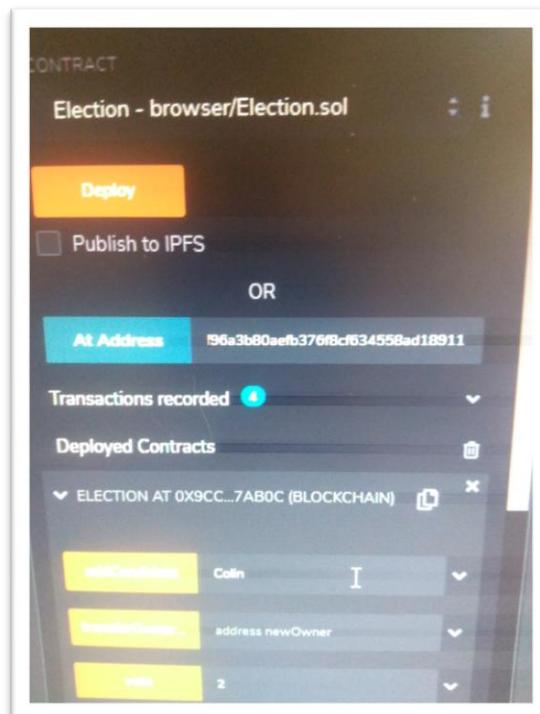
The screenshot shows the Etherscan interface for a Ropsten Testnet transaction. The transaction hash is 0x0171cc145ea188fa7c46ded296588d70b2157c517d894f91c553afa144a76c3a. The status is Success. It was included in block 8636618 with 3 block confirmations. The timestamp is 50 secs ago (Sep-07-2020 09:33:18 AM +UTC). The transaction originated from address 0xb9087a89c91071058ac85ffb37b675f0a78fc2cb and went to a contract at address 0x9cca507675d8a3e184766687889992a03527ab0c. The value sent was 0 Ether (\$0.00). The transaction fee was 0.0000994005 Ether (\$0.000000).

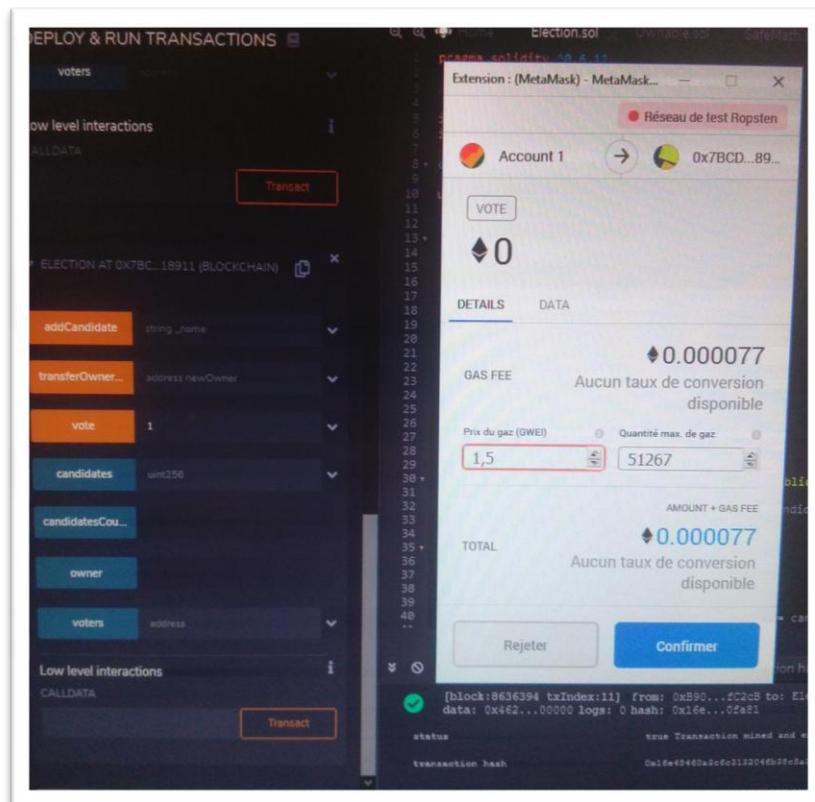
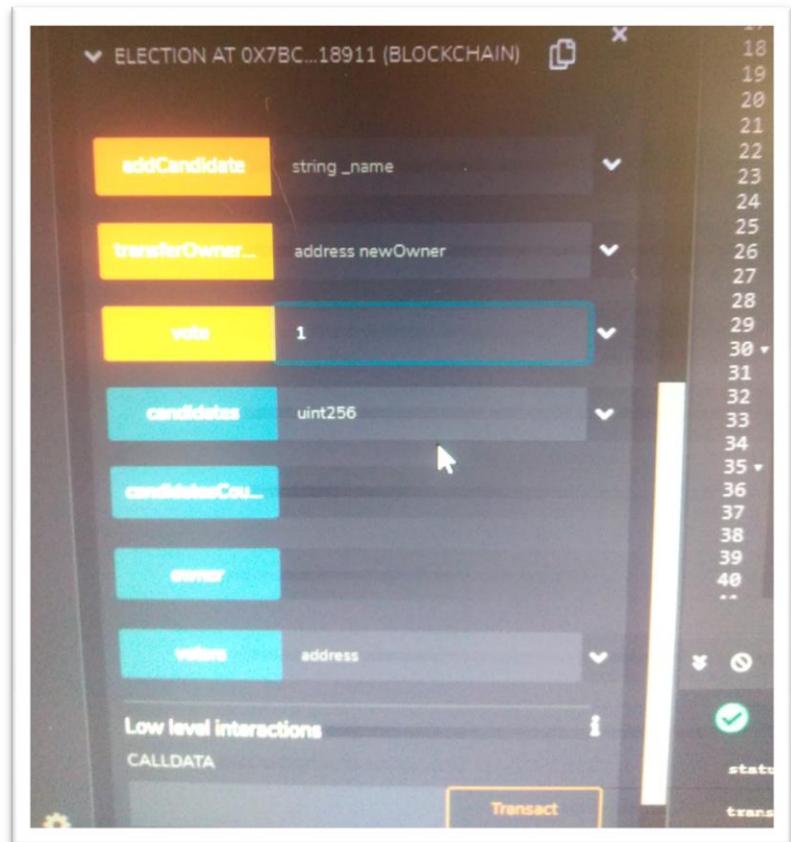
This screenshot shows the same transaction details as the first one, but with expanded logs. The logs section displays the function signature `Function: vote(uint256 proposal) ***` and the MethodID: `0x0121b93f`. The log data is shown as an array of hex values: `[0]: 0002`.

De même, pour ajouter un vote à l'utilisateur 2 :



Le but ensuite est d'envoyer des votes pour le contrat d'un camarade. Pour cela, on renseigne sa clé et procède de la même manière :





Résultat de transaction :

Etherscan

Ropsten Testnet Network

All Filters Search by Address

Transaction Details

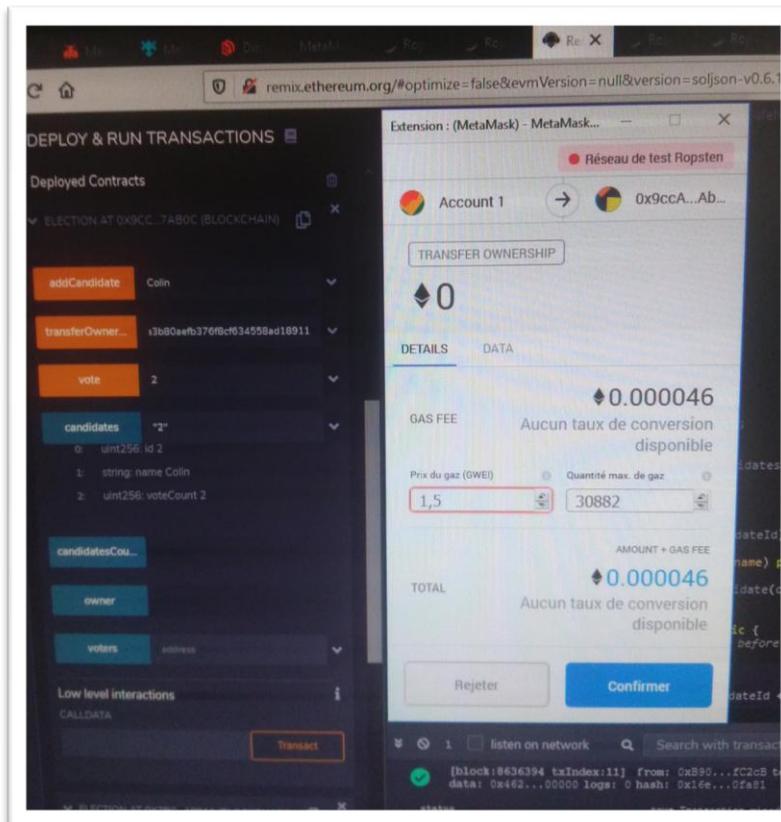
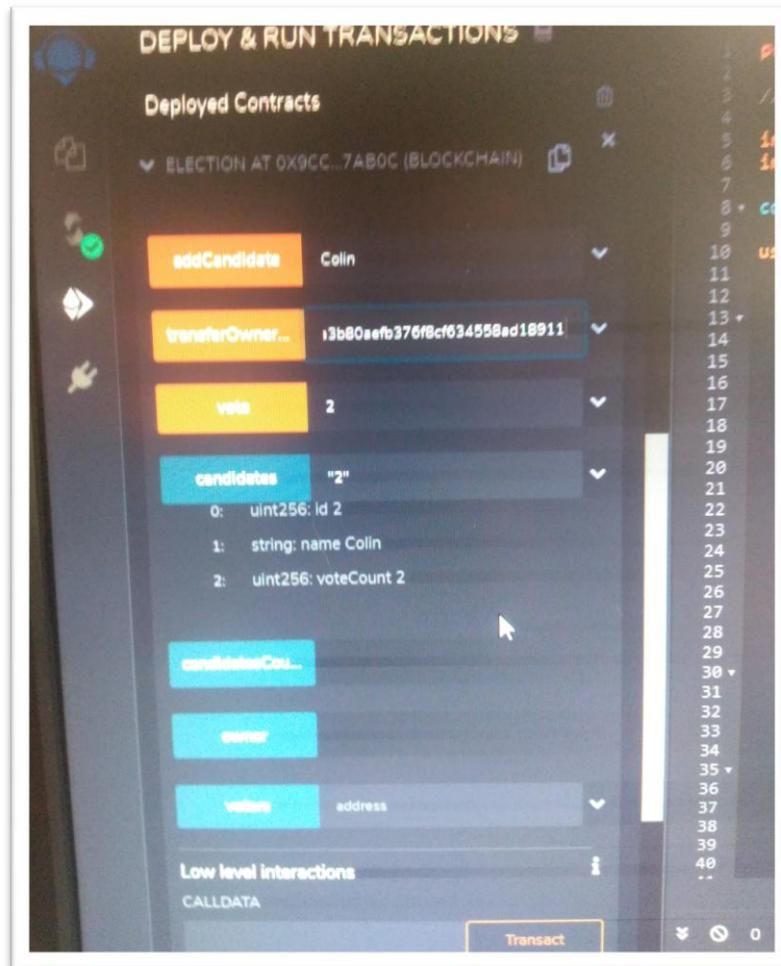
[Overview](#)

[This is a Ropsten Testnet transaction only]

⑦ Transaction Hash:	0x922bfe0602b5273f9d4391445395dba66e81fef2233f9a4a6cc751c615e2f3df Copy
⑦ Status:	Pending
⑦ Block:	(Pending)
⑦ Time Last Seen:	0 days 00 hr 00 min 19 secs ago (Sep-07-2020 09:49:18 AM)
⑦ From:	0xb9087a89c91071058ac85ffb37b675f0a78fc2cb Copy
⑦ Interacted With (To):	Contract 0x7bcd5c0bf96a3b80aefb376f8cf634558ad18911 Copy
⑦ Value:	0 Ether (\$0.000000)
⑦ Max Tnx Cost/Fee:	0.0000769005 Ether (\$0.000000)

[Click to see More](#) [↓](#)

Idem pour le deuxième candidat :



The screenshot shows the Etherscan.io interface for a Ropsten Testnet transaction. The transaction hash is 0x4b2a6ef7ea85afdf5aee06c777816eedff3c6a9e500221ef0bc8ea67733e14f7. The status is marked as 'Success'. The transaction was included in block 8636761, which has 7 block confirmations. It occurred 1 minute ago (Sep-07-2020 09:52:46 AM +UTC). The transaction originated from address 0xb9087a89c91071058ac85ffb37b675f0a78fc2cb and was sent to a contract at address 0x9cca507675d8a3e184766687889992a03527ab0c. The value transferred was 0 Ether (\$0.00). The transaction fee was 0.0000046323 Ether (\$0.000000). The gas limit was 30,882, and the gas used was 30,882 (100%). The gas price was 0.00000000015 Ether (1.5 Gwei). A note at the bottom indicates that the website uses cookies to improve your experience and has an updated Privacy Policy.

This screenshot shows the same transaction details as the previous one, but with a focus on the 'Logs (1)' tab. The input data for the transaction is shown as follows:

```

Function: transferOwnership(address newOwner) ***
MethodID: 0xf2fde38b
[0]: 000000000000000000000000000000007bcd5c0bf96a3b80aefb376f8cf634558ad18911

```

A 'View Input As' dropdown menu is visible below the input data. At the bottom, there is a link 'Click to see less'.

Enfin, si l'on veut pouvoir gérer les users en fonction du propriétaire du contrat :

The screenshot shows the Truffle UI interface. On the left, there's a sidebar with sections for ENVIRONMENT, Injected Web3, Ropsten (3) network, ACCOUNT (0xB90...fC2cB (3.997009039 ether)), GAS LIMIT (3000000), VALUE (0 wei), CONTRACT (Election - browser/Election.sol), Publish to IPFS, OR, At Address (0x6e3b80aefb370fbcf034558ad18911), Transactions recorded (0), and Deployed Contracts. The right side displays the Solidity code for the 'Election' contract.

```

import "./SafeMath.sol";
contract Election is Ownable {
    using SafeMath for uint256;
    struct Candidate {
        uint256 id;
        string name;
        uint voteCount;
    }
    mapping(address => bool) public voters;
    mapping(uint => Candidate) public candidates;
    uint public candidatesCount;
    event votedEvent ( uint indexed _candidateId);
    function addCandidate (string memory _name) public onlyOwner{
        candidatesCount++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    function vote (uint _candidateId) public {
        // require that they haven't voted before
        require(!voters[msg.sender]);
        // require a valid candidate
        require(_candidateId > 0 && _candidateId <= candidatesCount);
        // record that voter has voted
        voters[msg.sender] = true;
        // update candidate vote count
    }
}

```

Condition vérifiant que le owner du contrat est bien celui qui veut modifier les candidats.

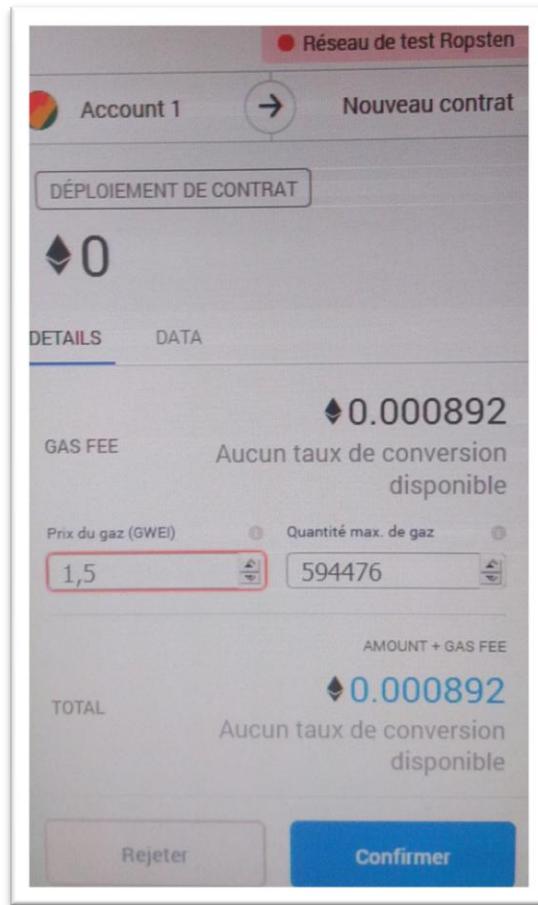
function addCandidate (string memory _name) public onlyOwner{

candidatesCount ++;

candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);

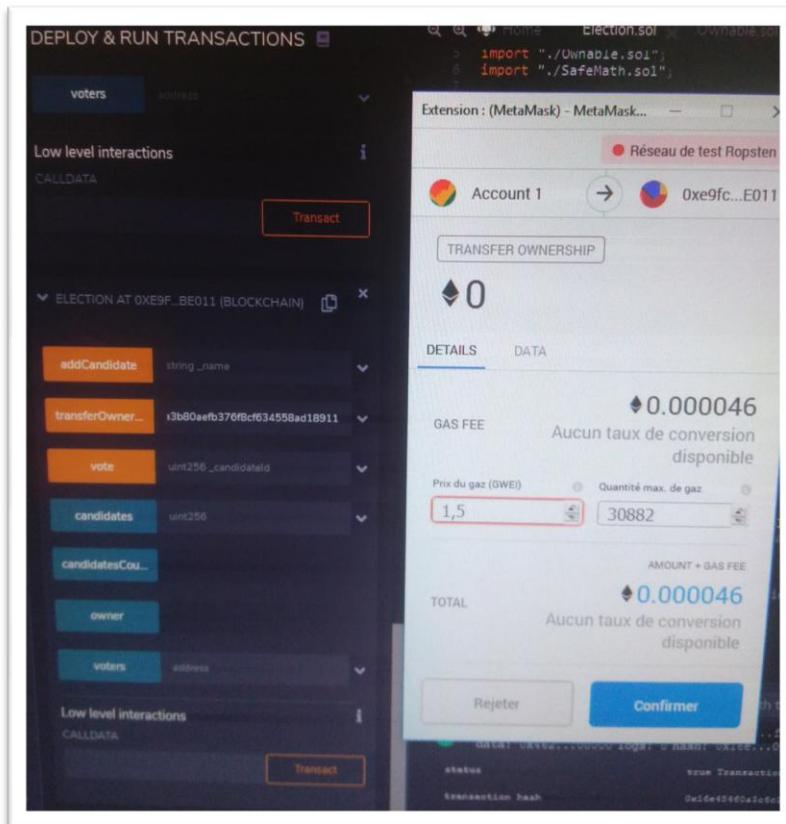
}

On n'a pas besoin de recompiler le code car on est en autocompile mais il faut cependant le redistribuer, il vient :



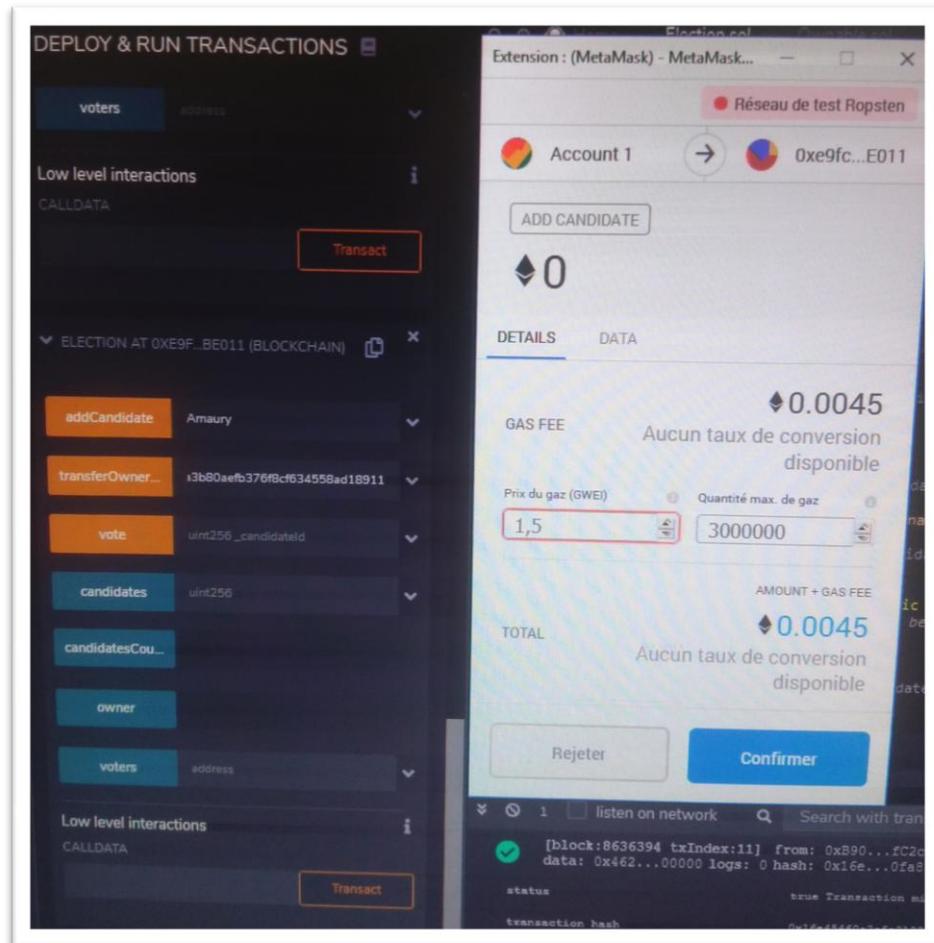
Résultat de distribution :

Enfin, nous avons vu comment changer le propriétaire d'un contrat :

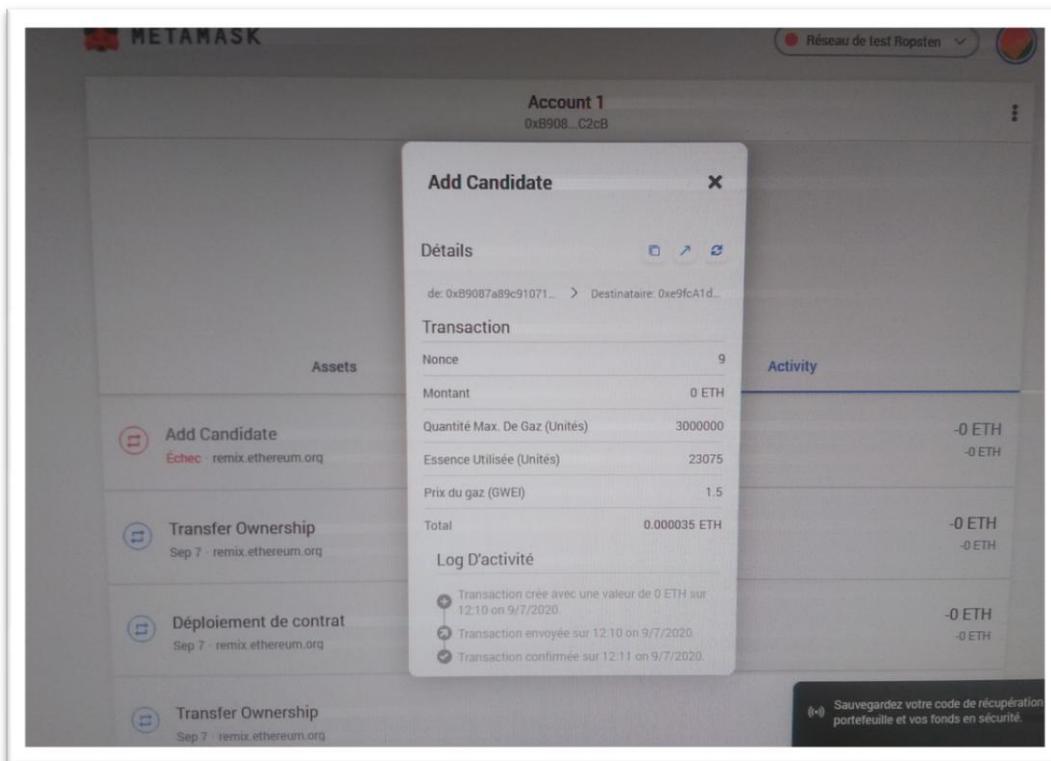


This screenshot shows the Etherscan transaction details page for the ownership transfer. The transaction hash is 0x60325b2ba2ae57ab7079bb33c049cd1619c42c1e81da0a13dcf34872cc538fc2. The status is 'Success' and it has 1 Block Confirmation. The transaction occurred 9 secs ago (Sep-07-2020 10:10:00 AM +UTC). It was sent from address 0xb9087a89c91071058ac85ffb37b675f0a78fc2cb to the contract address 0xe9fc1db53916e001d4cdb1f767f63a1b62be011. The value was 0 Ether (\$0.00). The transaction fee was 0.000046323 Ether (\$0.000000). The gas limit was 30.882, and the gas used by the transaction was 30.882 (100%). The gas price was 0.0000000015 Ether (1.5 Gwei). The nonce was 8. The input data shows the function call: transferOwnership(address newOwner) ***. MethodID: 0xf2fde30b. The input bytes are [0]: 000000000000000000000000000000007bcd5c0bf96a3b80aefb378fb3cfb34558ad18911.

Une fois effectué, on fait le test pour savoir si un utilisateur autre que le propriétaire du contrat peut créer un nouvel utilisateur :



On voit alors que la transaction à échoué :



Transaction Details

[Overview](#) [State](#)

[This is a Ropsten Testnet transaction only]

⑦ Transaction Hash:	0x35bd0e93188d7b564cc3df78696bc2764df7e83259b512fd9dbae721f12a51e	Copy
⑦ Status:	✖ Fail with error 'Not authorized operation'	
⑦ Block:	8636916	8 Block Confirmations
⑦ Timestamp:	① 1 min ago (Sep-07-2020 10:10:58 AM +UTC)	
⑦ From:	0xb9087a89c91071058ac85ff37b675f0a78fc2cb	
⑦ To:	Contract 0xe9fc1db53916e001d4cdb1f767f53a1b62be011 ⚠	
↳ Warning! Error encountered during contract execution [Reverted] ⓘ		
⑦ Value:	0 Ether (\$0.00)	
⑦ Transaction Fee:	0.0000346125 Ether (\$0.000000)	

[Click to see More](#) ↓