Assignment Tasks - Week1 - Group No. 5

This is a group activity for at least 3 students:

- Interact with "HelloWorld.sol" within your group to change message strings and change owners
- Write a report with each function execution and the transaction hash, if successful, or the revert reason, if failed
- Submit your weekend project by filling the form provided in Discord

Date of Submission

9-Aug-2024

REMIX CODE (LIVE EDITING)

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

```
/// @title EVM Bootcamp August 2024 Group-5 Week-1 Group Project

contract HelloWorld {
   /// @dev text is a private state variable to store any text
   string private text;
   /// @dev owner represents the owner of this Smart Contract
   address public owner;

   /// @dev onlyOwner modifier checks if the caller of the function
is the onwer/deployer of the Smart Contract
   /// @dev if not, then it raises an error with text "Caller is not
the owner"
   modifier onlyOwner()
   {
       require (msg.sender == owner, "Caller is not the owner");
       _;
   }

   /// @dev the address of the creator/deployer of this Smart
Contract
   /// @dev also, initializes the text variable
   constructor() {
       text = initialText();
       owner = msg.sender;
   }

   /// @dev initialText() returns a string, "Hello World"
   function initialText() public pure virtual returns (string memory)
{
       return "Hello World";
   }

   /// @dev helloWorld() function returns the state variable, 'text'
   function helloWorld() public view returns (string memory) {
       return text;
   }
```

```
    /// @dev only the deployer of the contract can change the state
variable, 'text'
    /// @dev setText() function sets the state variable, 'text'
    function setText(string calldata newText) public {
        require(msg.sender == owner);
        text = newText;
    }

    /// @dev this function changes the owner of the Smart Contract
    function transferOwnership(address newOwner) public onlyOwner {
        owner = newOwner;
    }

}
```

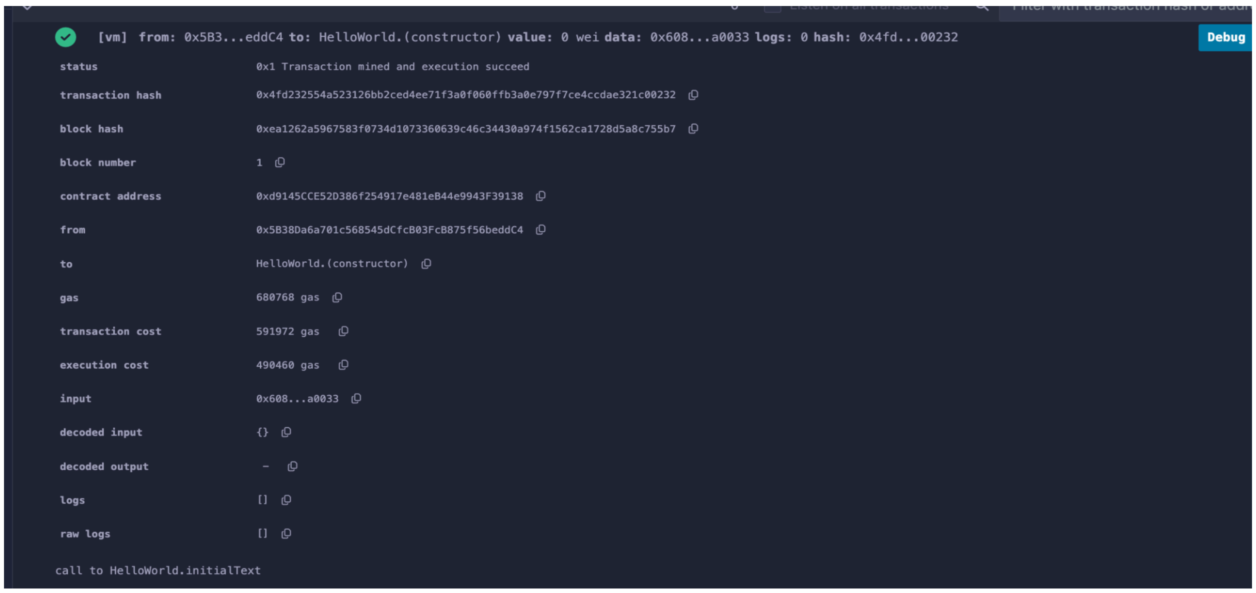
Report

Steps

1. Compiled the smart contract and then deployed

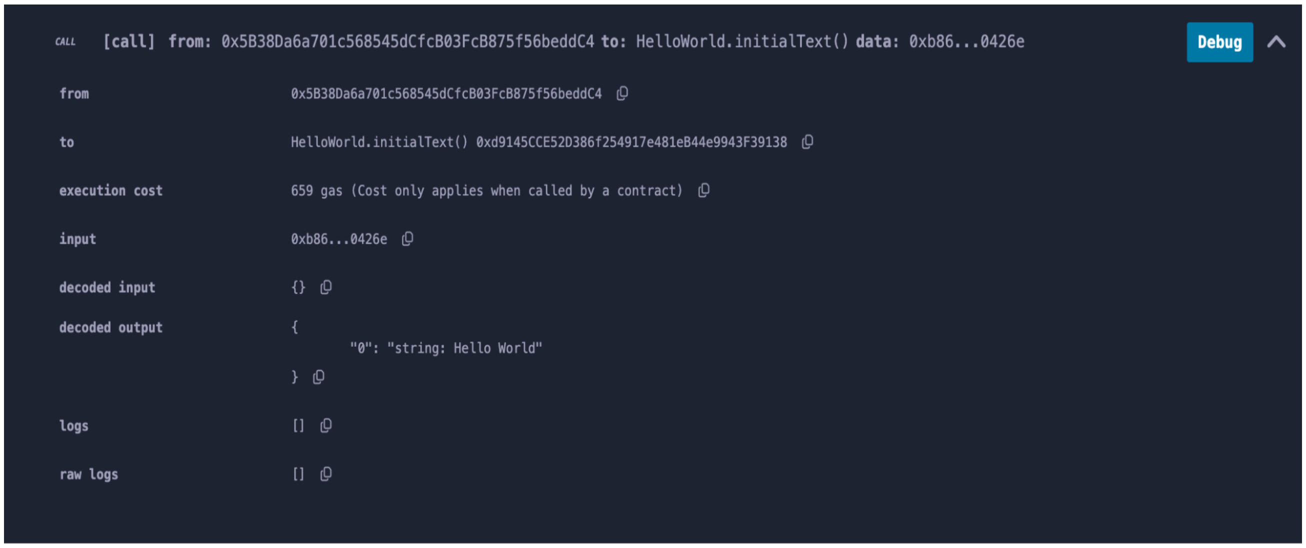
2. Constructor called on HelloWorld.sol deployment

   The constructor initializes the state variable, ‘text’ to “Hello World” and assigns the “owner” variable to the deployer.



```
[vm]  from: 0x5B3...eddC4 to: HelloWorld.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x4fd...00232
status                  0x1 Transaction mined and execution succeed
transaction hash        0x4fd232554a523126bb2ced4ee71f3a0f060ffb3a0e797f7ce4ccdae321c00232
block hash              0xea1262a5967583f0734d1073360639c46c34430a974f1562ca1728d5a8c755b7
block number            1
contract address        0xd9145CCE52D386f254917e481eB44e9943F39138
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      HelloWorld.(constructor)
gas                     680768 gas
transaction cost        591972 gas
execution cost          490460 gas
input                   0x608...a0033
decoded input           {}
decoded output           -
logs                    []
raw logs                []
call to HelloWorld.initialText
```

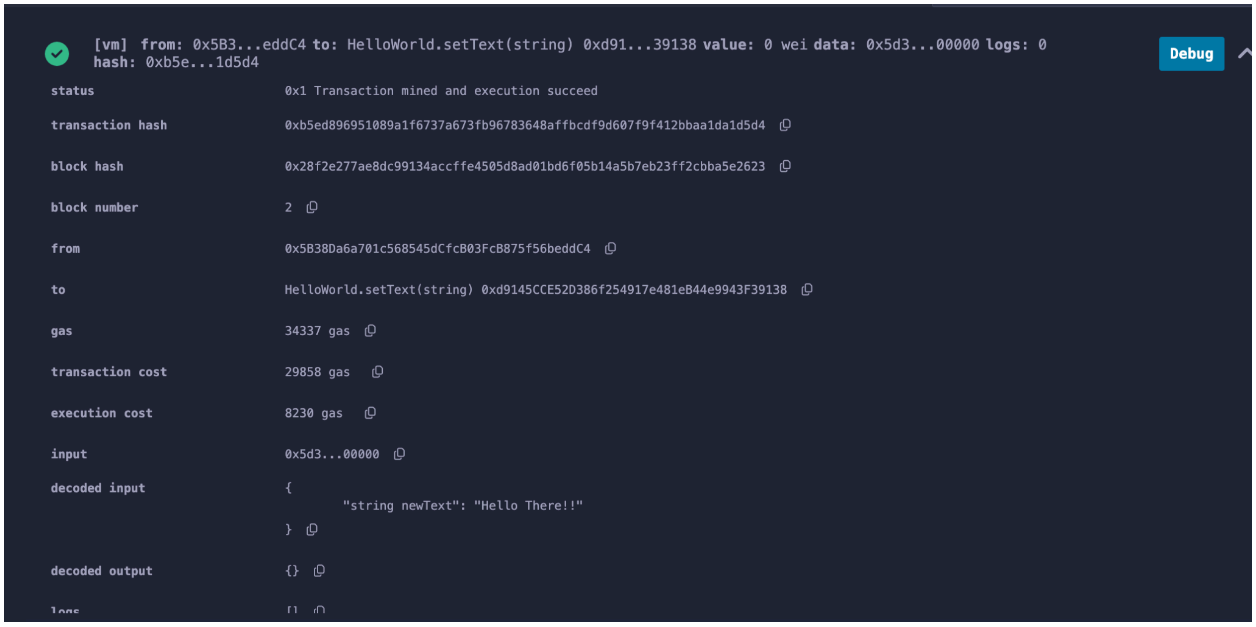
3. Calling HelloWorld.initialText()



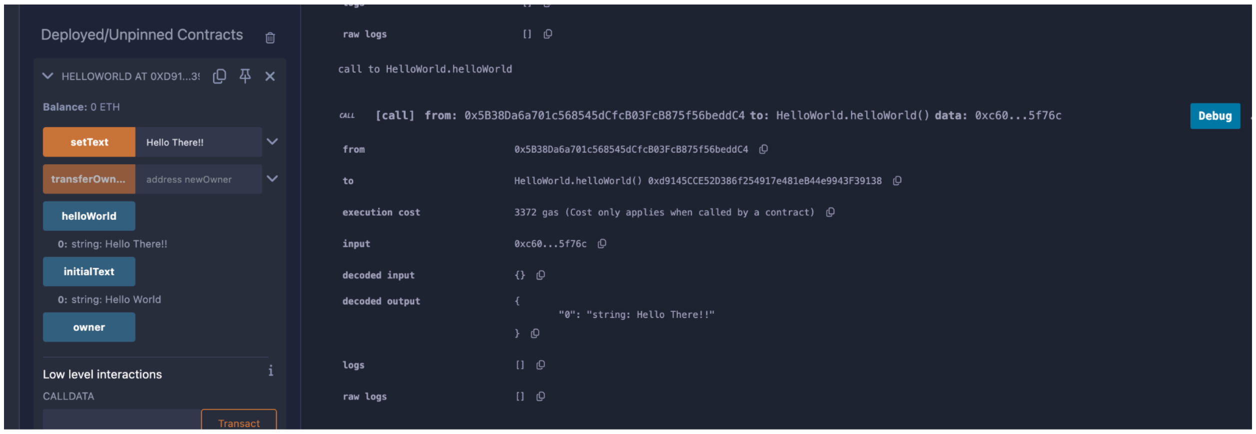
```
CALL   [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: HelloWorld.initialText() data: 0xb86...0426e
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      HelloWorld.initialText() 0xd9145CCE52D386f254917e481eB44e9943F39138
execution cost          659 gas (Cost only applies when called by a contract)
input                   0xb86...0426e
decoded input           {}
decoded output          {
                                "0": "string: Hello World"
                        }
logs                    []
raw logs                []
```

4. Calling HelloWorld.setText() function

Checking if the text was changed using the helloWorld()



```
[vm]  from: 0x5B3...eddC4 to: HelloWorld.setText(string) 0xd91...39138 value: 0 wei data: 0x5d3...00000 logs: 0
hash: 0xb5e...1d5d4
status                  0x1 Transaction mined and execution succeed
transaction hash        0xb5ed896951089a1f6737a673fb96783648affbcdf9d607f9f412bbaa1da1d5d4
block hash              0x28f2e277ae8dc99134accffe4505d8ad01bd6f05b14a5b7eb23ff2cbba5e2623
block number            2
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      HelloWorld.setText(string) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas                     34337 gas
transaction cost        29858 gas
execution cost          8230 gas
input                   0x5d3...00000
decoded input           {
                                "string newText": "Hello There!!"
                        }
decoded output          {}
logs                    []
```



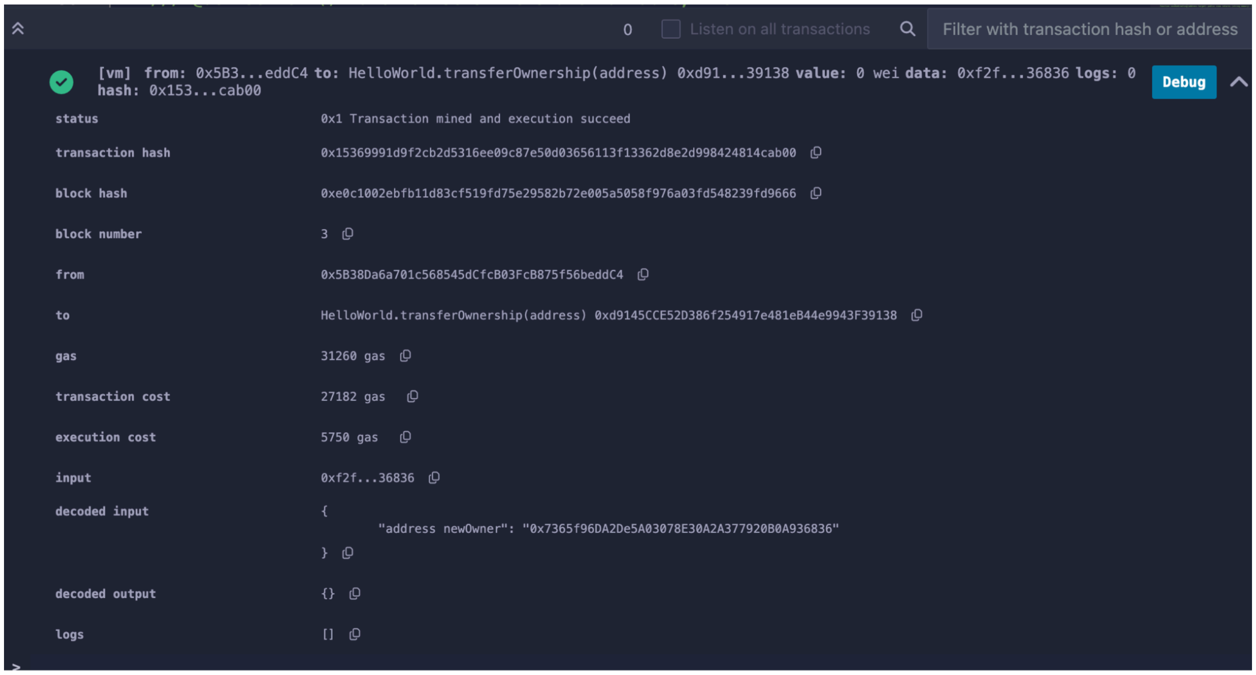
```
raw logs                []

call to HelloWorld.helloWorld

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: HelloWorld.helloWorld() data: 0xc60...5f76c
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      HelloWorld.helloWorld() 0xd9145CCE52D386f254917e481eB44e9943F39138
execution cost          3372 gas (Cost only applies when called by a contract)
input                   0xc60...5f76c
decoded input           {}
decoded output          {
                                "0": "string: Hello There!!"
                        }
logs                    []
raw logs                []
```

5. Calling transferOwnership()

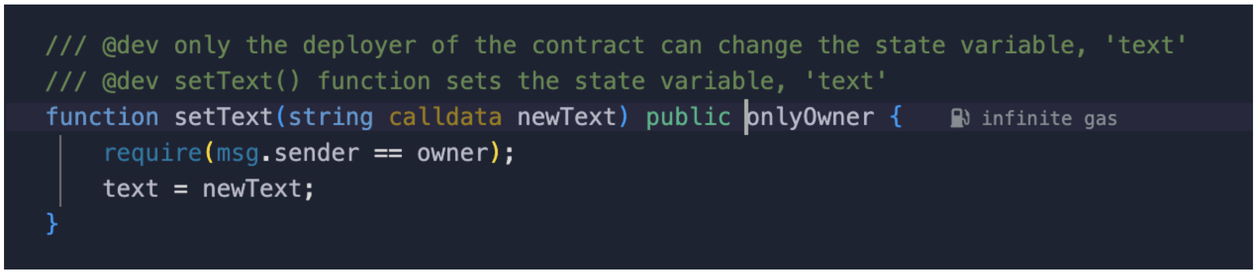
   Contract ownership transferred to a new address. We verified this by checking the owner public variable



```
[vm] from: 0x5B3...eddC4 to: HelloWorld.transferOwnership(address) 0xd91...39138 value: 0 wei data: 0xf2f...36836 logs: 0
hash: 0x153...cab00
status              0x1 Transaction mined and execution succeed
transaction hash    0x15369991d9f2cb2d5316ee09c87e50d03656113f13362d8e2d998424814cab00
block hash          0xe0c1002ebfb11d83cf519fd75e29582b72e005a5058f976a03fd548239fd9666
block number        3
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  HelloWorld.transferOwnership(address) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas                 31260 gas
transaction cost    27182 gas
execution cost      5750 gas
input               0xf2f...36836
decoded input       {
                           "address newOwner": "0x7365f96DA2De5A03078E30A2A377920B0A936836"
                    }
decoded output      {}
logs                []
```

0
Listen on all transactions
Filter with transaction hash or address
Debug

6. Adding onlyOwner modifier to setText() function

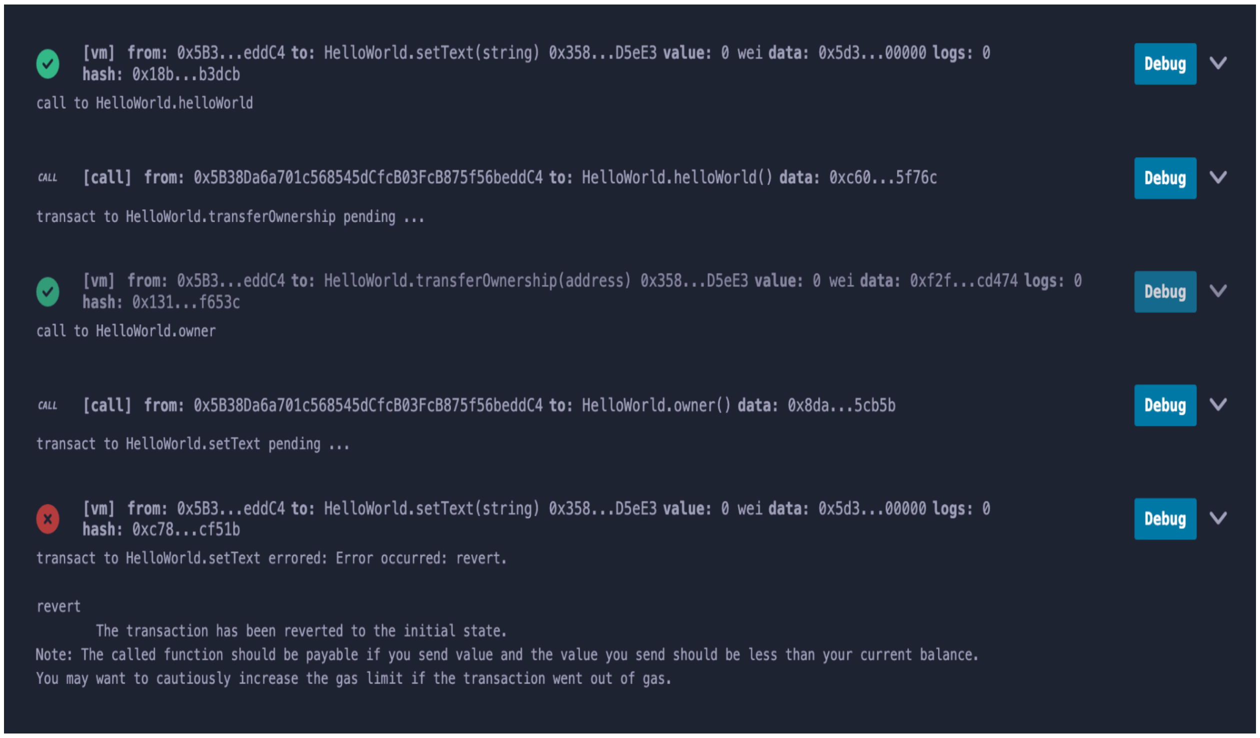


```
/// @dev only the deployer of the contract can change the state variable, 'text'
/// @dev setText() function sets the state variable, 'text'
function setText(string calldata newText) public onlyOwner {    infinite gas
    require(msg.sender == owner);
    text = newText;
}
```

7. Add onlyOwner modifier to setText() function and test how it behaves

- Called setText() function as a owner/deployer and changed text - WORKED
- Checked if the text changed using helloWorld() function - WORKED
- Changed ownership using transferOwnership() function - WORKED
- Checked if the ownership changed using owner state variable - WORKED

- Called setText() function and tried to change the text - ERROR!
  (I'm not the owner anymore because the ownership is transferred to different address)



```
✓ [vm] from: 0x5B3...eddC4 to: HelloWorld.setText(string) 0x358...D5eE3 value: 0 wei data: 0x5d3...00000 logs: 0
  hash: 0x18b...b3dcb
call to HelloWorld.helloWorld

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: HelloWorld.helloWorld() data: 0xc60...5f76c

transact to HelloWorld.transferOwnership pending ...

✓ [vm] from: 0x5B3...eddC4 to: HelloWorld.transferOwnership(address) 0x358...D5eE3 value: 0 wei data: 0xf2f...cd474 logs: 0
  hash: 0x131...f653c
call to HelloWorld.owner

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: HelloWorld.owner() data: 0x8da...5cb5b

transact to HelloWorld.setText pending ...

✗ [vm] from: 0x5B3...eddC4 to: HelloWorld.setText(string) 0x358...D5eE3 value: 0 wei data: 0x5d3...00000 logs: 0
  hash: 0xc78...cf51b
transact to HelloWorld.setText errored: Error occurred: revert.

revert
        The transaction has been reverted to the initial state.
Note: The called function should be payable if you send value and the value you send should be less than your current balance.
You may want to cautiously increase the gas limit if the transaction went out of gas.
```

-------- Fin. --------