

Practical 10
Blockchain Technology
2CSDE93

Dhruv Sonani
20BCE527

Date
November 30, 2022



Department of Computer Science and Engineering
Institute of Technology
Nirma University
Ahmedabad

Aim: Implementing Tic-Tac-Toe game

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.7.0 < 0.9.0;

contract TicTacToe
{
    address player1;
    address player2;
    string[3][3] board;
    address whosTurn;
    bool gameOver;

    constructor(address _player1,address _player2){
        player1 = _player1;
        player2 = _player2;
        whosTurn = _player1;
        gameOver = false;
    }

    modifier _playerOnly() {
        require(msg.sender == whosTurn);
        _;
    }

    function Move(uint256 x,uint256 y)public _playerOnly
    {
        require(x <= 2 && x >= 0 && y >= 0 && y <= 2 &&
bytes(board[x][y]).length != 0,"Please enter a valid index");
        require(!gameOver,"Game over");

        if(whosTurn==player1)
        {
            board[x][y] = 'X'; whosTurn = player2;
        }
        else
        {
            board[x][y] = 'O'; whosTurn = player1;
        }

        string memory checkOver = checkGameOver();
    }
}
```

```

        if(keccak256(bytes(checkOver)) == keccak256(bytes("0")) ||
keccak256(bytes(checkOver)) == keccak256(bytes("X")))
        {
            gameOver = true;
        }
    }

function checkGameOver() public view returns (string memory)
{
    for(uint256 i=0;i<board.length;i++)
    {
        for(uint256 k=1;k<board[i].length;k++)
        {
            if((keccak256(abi.encodePacked((board[i][k]))) !=
(keccak256(abi.encodePacked((board[i][ 0])))))
            {
                break;
            }
            else if(k == board[i].length -1)
            {
                return board[i][0];
            }
        }
    }

    for(uint256 i=1;i<board.length;i++)
    {
        if((keccak256(abi.encodePacked((board[i][i]))) !=
(keccak256(abi.encodePacked((board[0][ 0])))))
        {
            break;
        }

        if(i == board.length - 1)
        {
            return board[0][0];
        }
    }

    uint256 tmp = board.length -1;

    for(uint256 i=0;i<board.length;i++)
    {
        if((keccak256(abi.encodePacked((board[i][tmp]))) !=
(keccak256(abi.encodePacked((board[0] [board.length - 1])))))
        {
            break;

```

```
    }

    tmp = tmp - 1;

    if(i == board.length - 1)
    {
        return board[0][board.length - 1];
    }
}

return "Not Over";
}
```

Output:

The screenshot shows the 'DEPLOY & RUN TRANSACTIONS' interface in the Remix IDE. The left sidebar contains icons for environment selection, search, account management, gas limit, value, and contract selection. The main panel is titled 'DEPLOY & RUN TRANSACTIONS' with a green checkmark and a right arrow. It includes sections for 'ENVIRONMENT' (Remix VM (London)), 'ACCOUNT' (0x5B3...eddC4 (99.999999%)), 'GAS LIMIT' (3000000), 'VALUE' (0 Wei), and 'CONTRACT' (TicTacToe - contracts/Tic_Tac_Toe.sol). The 'DEPLOY' section shows two player addresses, a 'transact' button, and a 'Publish to IPFS' checkbox. Below this is an 'OR' section with 'At Address' and 'Load contract from Address' options. The bottom status bar indicates 'Transactions recorded 1'.

DEPLOY & RUN TRANSACTIONS ✓ >

ENVIRONMENT ⓘ

Remix VM (London) ⓘ

VM

ACCOUNT ⓘ

0x5B3...eddC4 (99.999999% ⓘ ⓘ

GAS LIMIT

3000000

VALUE

0 Wei ⓘ

CONTRACT (Compiled by Remix)

TicTacToe - contracts/Tic_Tac_Toe.sol ⓘ

DEPLOY ^

_PLAYER1: "0x5B38Da6a701c568545dCfcB"

_PLAYER2: "0xAb8483F64d9C6d1EcF9b849"

ⓘ Calldata ⓘ Parameters **transact**

☐ Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 1 ⓘ >



DEPLOY & RUN TRANSACTIONS ✓ >



CONTRACT (Compiled by Remix)



TicTacToe - contracts/Tic_Tac_Toe.sol



Deploy

0x5B38Da6a701c568545dCfc



☐ Publish to IPFS



OR



At Address

Load contract from Address



Transactions recorded 1 >

Deployed Contracts



▼ TICTACTOE AT 0XD91...39138 (MEI) ×

Balance: 0 ETH

Move

uint256 x, uint256 y



checkGam...

Low level interactions



CALLDATA

Transact

