Homework-1: Programming a Faucet contract on the Sepolia testnet using Remix

110321006 蔡秉翰

110321007 劉竑毅

我們的這個智能合約是一個簡單的水龍頭（Faucet）合約，用於分發以太幣給其他地址。以下是程式的結構及流程說明：

合約擁有者（Owner）: 合約中有一個 owner 變數，用來記錄合約的擁有者地址。擁有者是在合約部署時設置的，一旦設置後便無法更改。

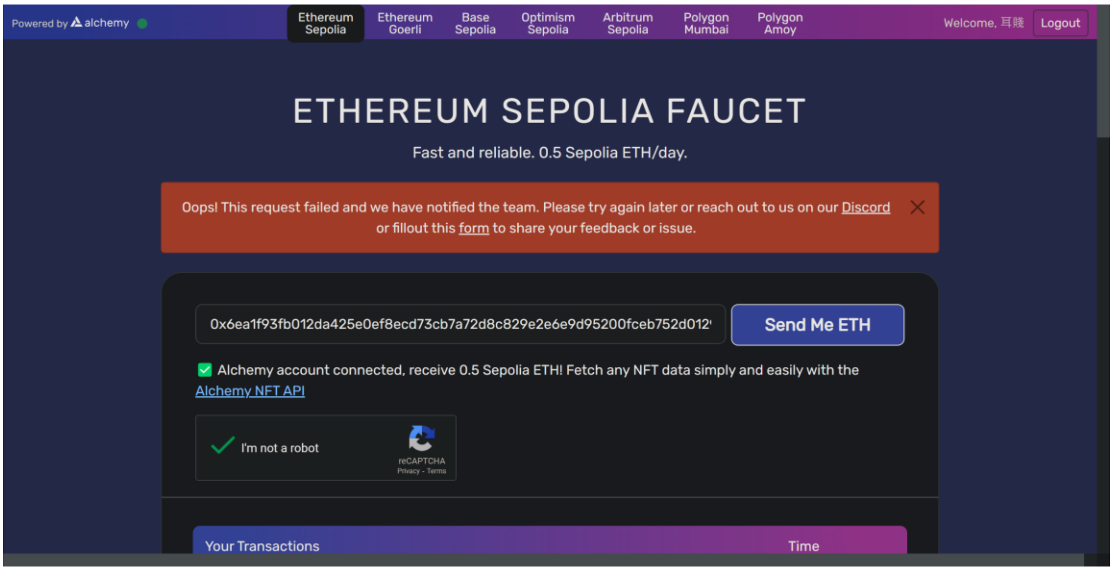
接收以太幣的函式: 合約中有一個 receive 函式，它用來接收以太幣轉入合約的操作。這個函式使用了 external 可見性，表示它只能被外部調用，而不能在合約內部被調用。

提取以太幣的函式: withdraw 函式允許合約的外部地址提取以太幣。提取的金額由函式的參數指定。在提取前，會檢查合約的餘額是否足夠。

存入以太幣的函式: deposit 函式用來向合約存入以太幣。存入的金額由函式調用時所傳入的以太幣數量指定。在存入時，會檢查傳入的金額是否大於零。

合約管理者提取所有資金: withdrawAll 函式允許合約的擁有者提取合約內所有的以太幣。這是一個管理功能，只有合約的擁有者才有權限執行這個操作。

這些函式一起組成了一個簡單的水龍頭合約，允許存款、提款和管理合約內的資金。



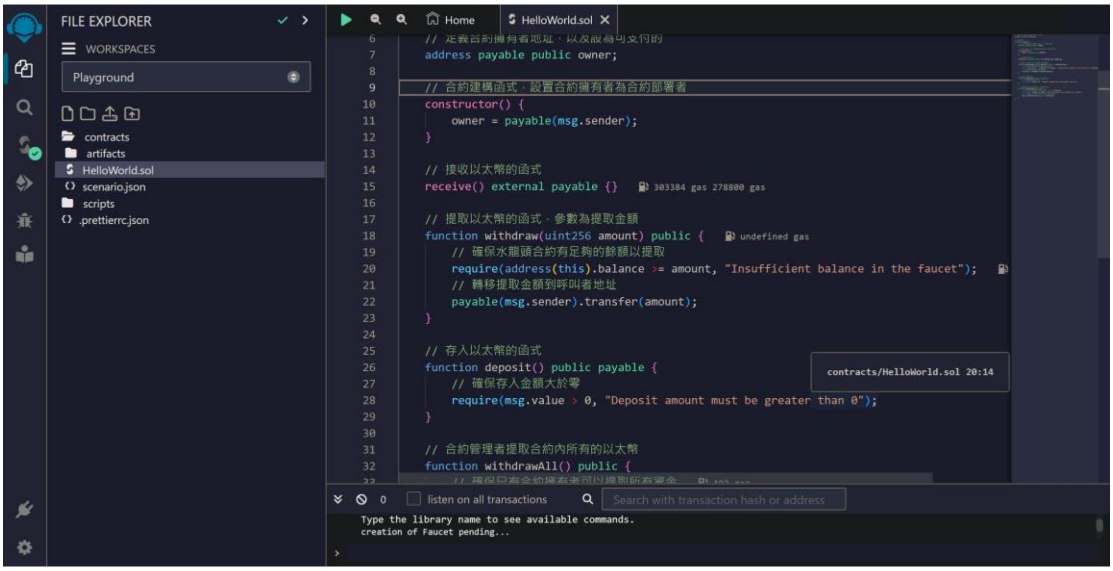
ETHEREUM SEPOLIA FAUCET

Fast and reliable. 0.5 Sepolia ETH/day.

Oops! This request failed and we have notified the team. Please try again later or reach out to us on our Discord or fillout this form to share your feedback or issue.

0x6ea1f93fb012da425e0ef8ecd73cb7a72d8c829e2e6e9d95200fceb752d012

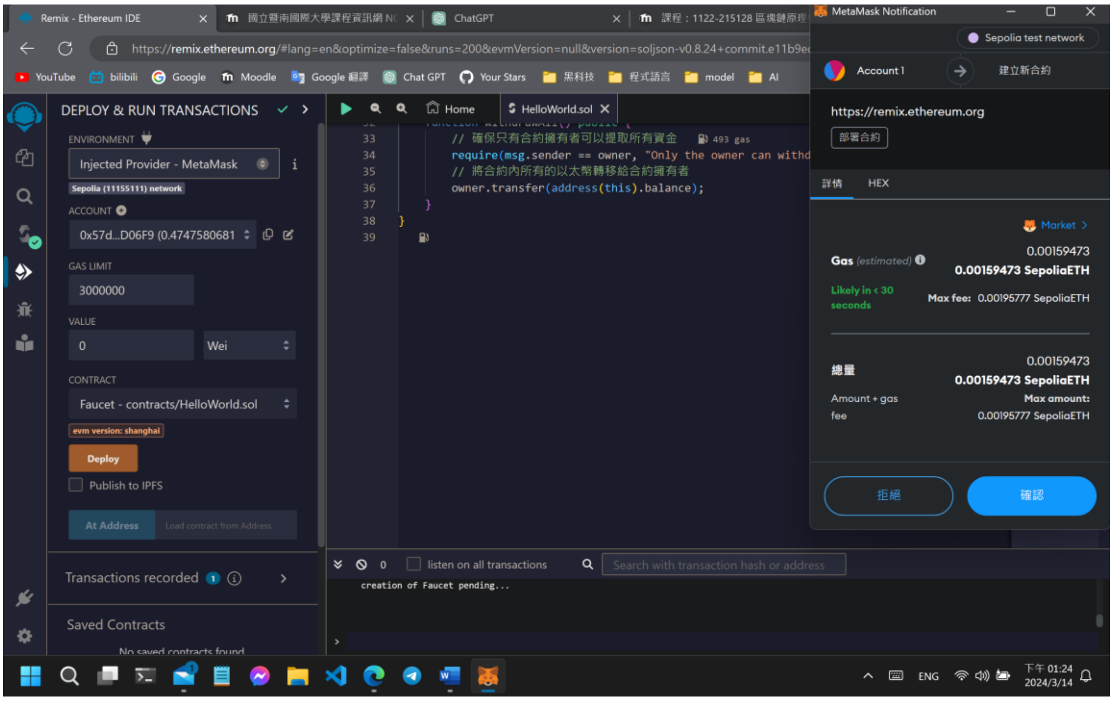
☑ Alchemy account connected, receive 0.5 Sepolia ETH! Fetch any NFT data simply and easily with the Alchemy NFT API



```
    // 定義合約擁有者地址，以及設為可支付的
    address payable public owner;

    // 合約建構函式，設置合約擁有者為合約部署者
    constructor() {
        owner = payable(msg.sender);
    }

    // 接收以太幣的函式
    receive() external payable {}   303384 gas 278800 gas

    // 提取以太幣的函式，參數為提取金額
    function withdraw(uint256 amount) public {   undefined gas
        // 確保水龍頭合約有足夠的餘額以提取
        require(address(this).balance >= amount, "Insufficient balance in the faucet");
        // 轉移提取金額到呼叫者地址
        payable(msg.sender).transfer(amount);
    }

    // 存入以太幣的函式
    function deposit() public payable {
        // 確保存入金額大於零
        require(msg.value > 0, "Deposit amount must be greater than 0");
    }

    // 合約管理者提取合約內所有的以太幣
    function withdrawAll() public {
```

contracts/HelloWorld.sol 20:14
0
listen on all transactions
Search with transaction hash or address

```
Type the library name to see available commands.
creation of Faucet pending...
```



https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.24+commit.e11b9ec

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0x57d...D06F9 (0.4747580681

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: Faucet - contracts/HelloWorld.sol

evm version: shanghai

```
        // 確保只有合約擁有者可以提取所有資金   493 gas
        require(msg.sender == owner, "Only the owner can withd
        // 將合約內所有的以太幣轉移給合約擁有者
        owner.transfer(address(this).balance);
    }
}
```

0
listen on all transactions
Search with transaction hash or address

```
creation of Faucet pending...
```

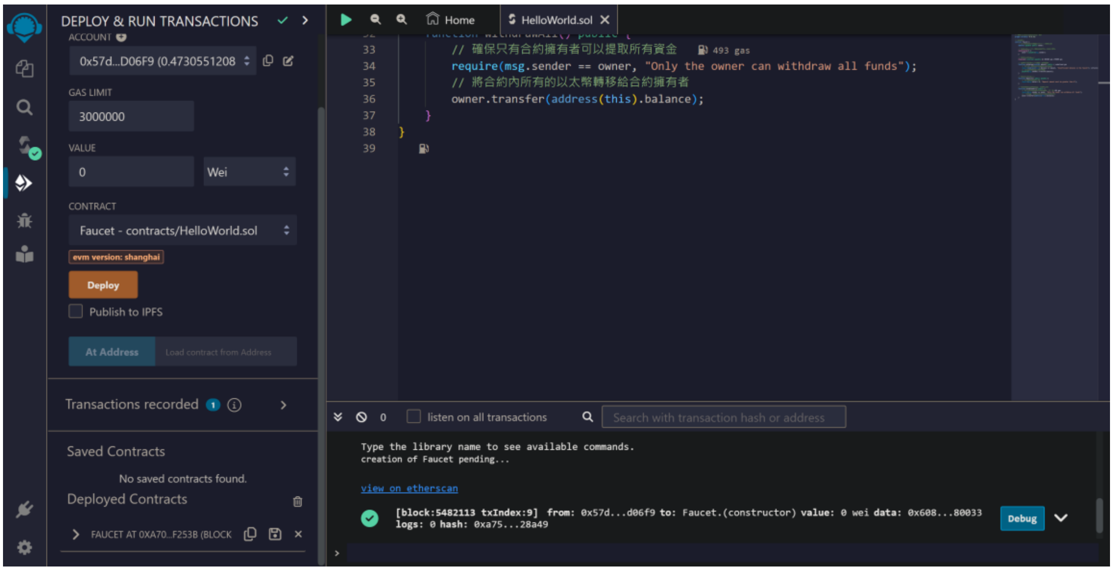
https://remix.ethereum.org

Gas (estimated) 0.00159473 **0.00159473 SepoliaETH**

Likely in < 30 seconds — Max fee: 0.00195777 SepoliaETH

總量 0.00159473 **0.00159473 SepoliaETH**

Amount + gas fee — Max amount: 0.00195777 SepoliaETH



ACCOUNT
0x57d...D06F9 (0.4730551208

GAS LIMIT
3000000

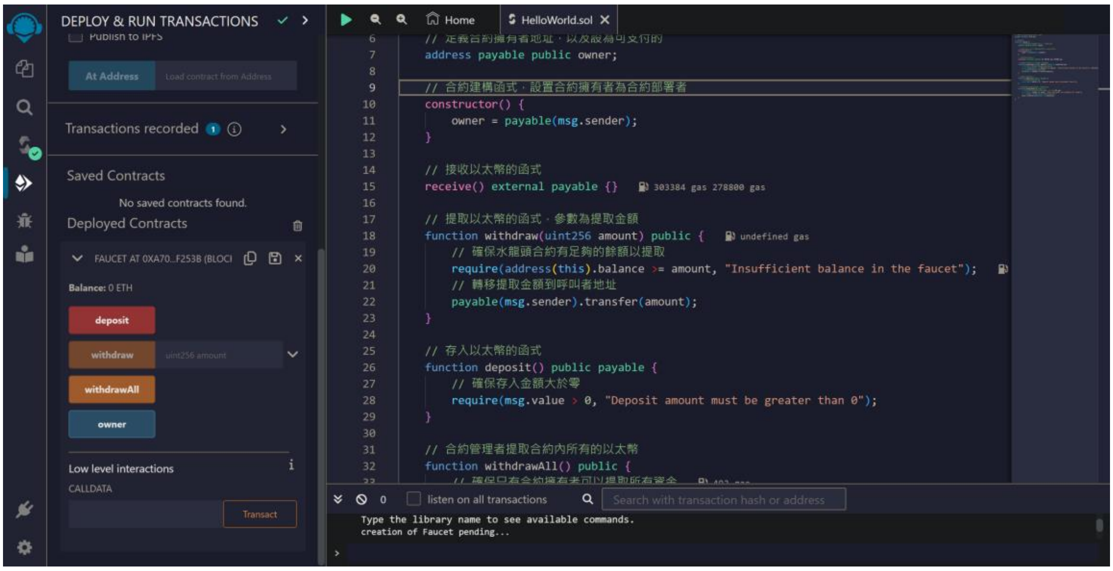
VALUE
0 Wei

CONTRACT
Faucet - contracts/HelloWorld.sol

No saved contracts found.

```
        // 確保只有合約擁有者可以提取所有資金    493 gas
        require(msg.sender == owner, "Only the owner can withdraw all funds");
        // 將合約內所有的以太幣轉移給合約擁有者
        owner.transfer(address(this).balance);
    }
}
```

```
Type the library name to see available commands.
creation of Faucet pending...

view on etherscan

[block:5482113 txIndex:9] from: 0x57d...d06f9 to: Faucet.(constructor) value: 0 wei data: 0x608...80033
logs: 0 hash: 0xa75...28a49
```

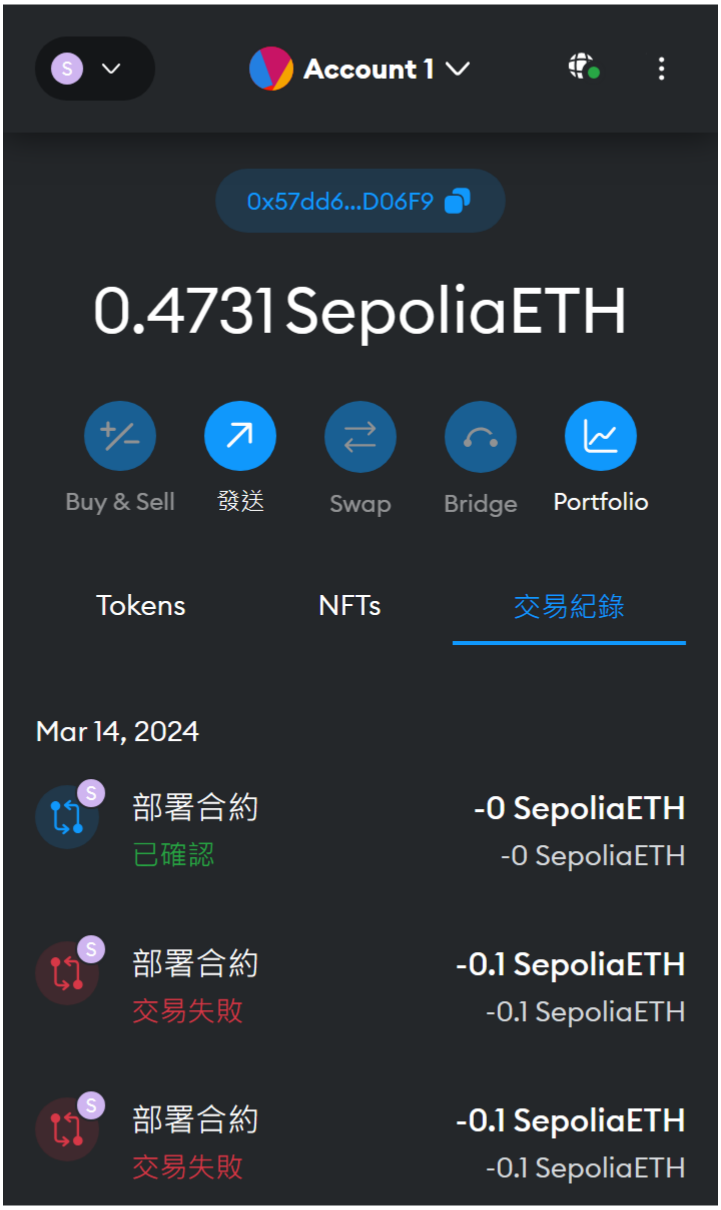


No saved contracts found.

Balance: 0 ETH

```
    // 定義合約擁有者地址，以及設為可支付的
    address payable public owner;

    // 合約建構函式，設置合約擁有者為合約部署者
    constructor() {
        owner = payable(msg.sender);
    }

    // 接收以太幣的函式
    receive() external payable {}    303384 gas 278800 gas

    // 提取以太幣的函式，參數為提取金額
    function withdraw(uint256 amount) public {    undefined gas
        // 確保水龍頭合約有足夠的餘額以提取
        require(address(this).balance >= amount, "Insufficient balance in the faucet");
        // 轉移提取金額到呼叫者地址
        payable(msg.sender).transfer(amount);
    }

    // 存入以太幣的函式
    function deposit() public payable {
        // 確保存入金額大於零
        require(msg.value > 0, "Deposit amount must be greater than 0");
    }

    // 合約管理者提取合約內所有的以太幣
    function withdrawAll() public {
```

```
Type the library name to see available commands.
creation of Faucet pending...
```



0.4731 SepoliaETH

Mar 14, 2024

部署合約 -0 SepoliaETH
已確認 -0 SepoliaETH

部署合約 -0.1 SepoliaETH
交易失敗 -0.1 SepoliaETH

部署合約 -0.1 SepoliaETH
交易失敗 -0.1 SepoliaETH

在使用 Remix IDE 時，遇到的兩個警告是關於缺少 SPDX license identifier 和未指定所需的編譯器版本。

解決這些問題的方法如下：

添加 SPDX license identifier：在程式碼的開頭添加// SPDX-License-Identifier: MIT 註釋，表示程式碼採用 MIT 許可證發布。

指定所需的編譯器版本：在程式碼的開頭添加 pragma solidity ^0.8.0; 或 pragma solidity ^0.8.24