REMIX

THE SOLIDITY IDE IN THE CLOUD

GOOD TO KNOW

- Previously Browser-Solidity
- For Smart Contract Development
  - Integrated Debugger
  - Integrated Testing Environment
  - Static Analysis
- Can directly publish on a GIST

- Can also work inside MIST or with MetaMask
  - Can directly deploy

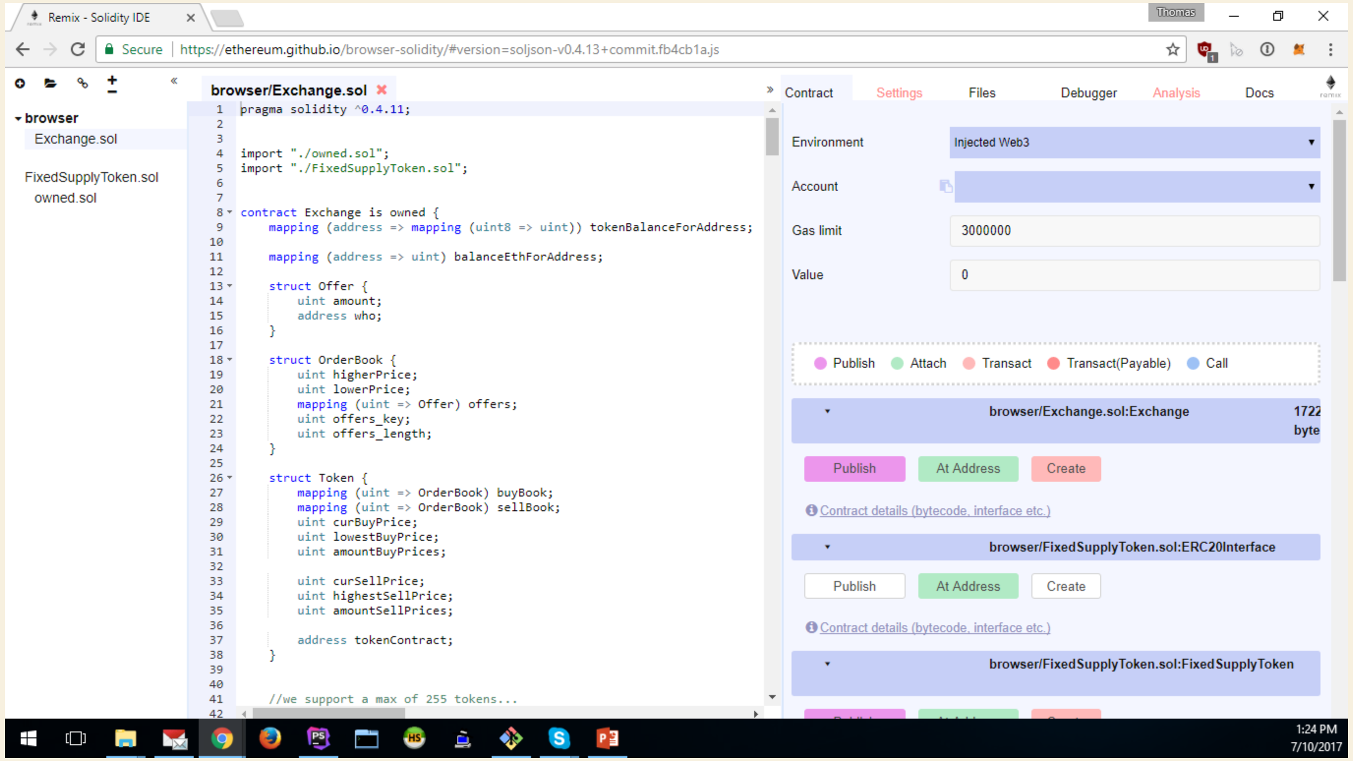
- Available on http://remix.ethereum.org

DEVELOPMENT

- Syntax highlighting
- Multiple Files
- Error Highlighting (integrated static analysis)
- Simulated Blockchain (via VM)
  - Can also work with injected web3 object
  - Or with a http-rpc (like a local geth node)

- Deploy contracts
- Watch details (bytecode, ABI)

DEVELOPMENT



Remix - Solidity IDE
Secure
https://ethereum.github.io/browser-solidity/#version=soljson-v0.4.13+commit.fb4cb1a.js
Thomas
browser
Exchange.sol
FixedSupplyToken.sol
owned.sol
browser/Exchange.sol
Contract
Settings
Files
Debugger
Analysis
Docs
Publish
Attach
Transact
Transact(Payable)
Call
browser/Exchange.sol:Exchange
1722 byte
Publish
At Address
Create
Contract details (bytecode, interface etc.)
browser/FixedSupplyToken.sol:ERC20Interface
Publish
At Address
Create
Contract details (bytecode, interface etc.)
browser/FixedSupplyToken.sol:FixedSupplyToken
1:24 PM
7/10/2017

```
pragma solidity ^0.4.11;


import "./owned.sol";
import "./FixedSupplyToken.sol";


contract Exchange is owned {
    mapping (address => mapping (uint8 => uint)) tokenBalanceForAddress;

    mapping (address => uint) balanceEthForAddress;

    struct Offer {
        uint amount;
        address who;
    }

    struct OrderBook {
        uint higherPrice;
        uint lowerPrice;
        mapping (uint => Offer) offers;
        uint offers_key;
        uint offers_length;
    }

    struct Token {
        mapping (uint => OrderBook) buyBook;
        mapping (uint => OrderBook) sellBook;
        uint curBuyPrice;
        uint lowestBuyPrice;
        uint amountBuyPrices;

        uint curSellPrice;
        uint highestSellPrice;
        uint amountSellPrices;

        address tokenContract;
    }


    //we support a max of 255 tokens...
```

Environment: Injected Web3

Account:

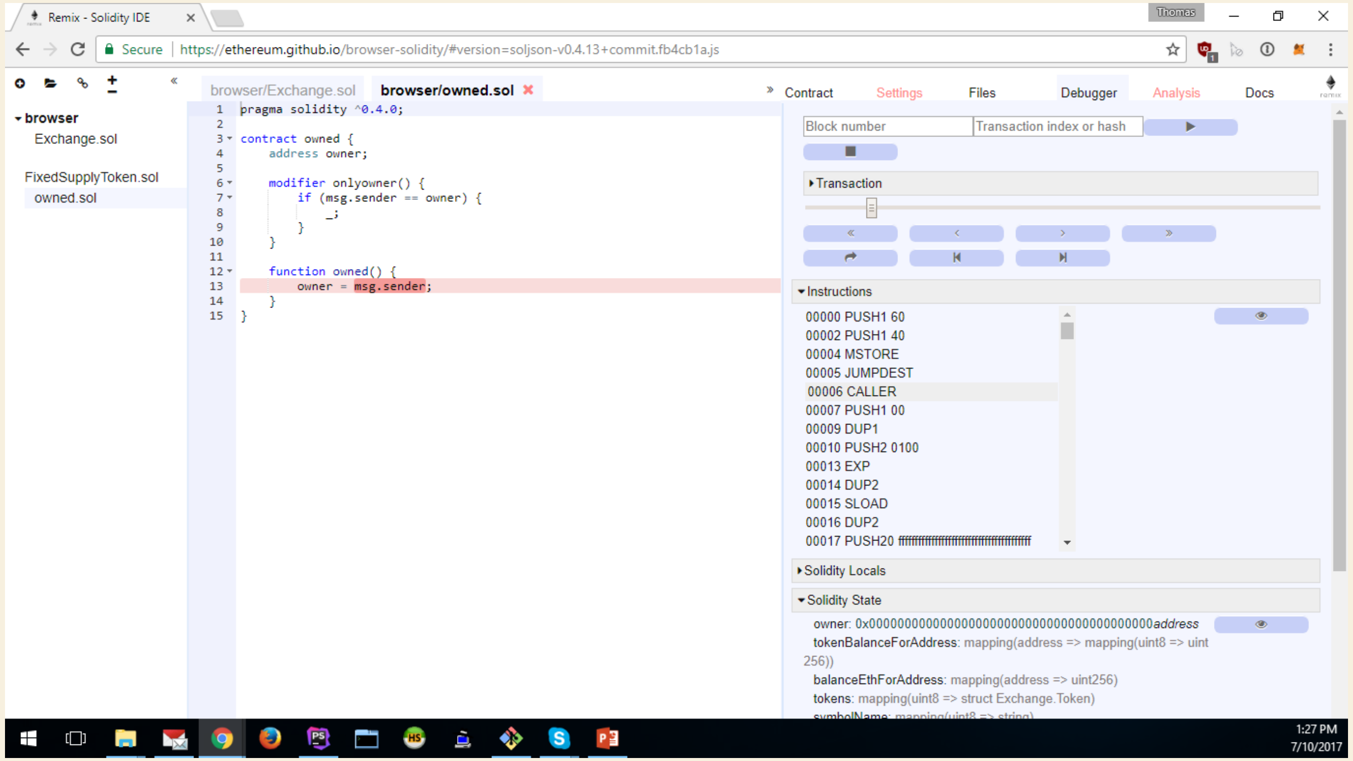
Gas limit: 3000000

Value: 0

DEBUGGING

- Step by Step
  - With Code highlighting
- View Instruction
- Monitor variables
  - Locals & State
- Call Data
- Call Stack

DEBUGGING



```
pragma solidity ^0.4.0;

contract owned {
    address owner;

    modifier onlyowner() {
        if (msg.sender == owner) {
            _;
        }
    }

    function owned() {
        owner = msg.sender;
    }
}
```

```
00000 PUSH1 60
00002 PUSH1 40
00004 MSTORE
00005 JUMPDEST
00006 CALLER
00007 PUSH1 00
00009 DUP1
00010 PUSH2 0100
00013 EXP
00014 DUP2
00015 SLOAD
00016 DUP2
00017 PUSH20 ffffffffffffffffffffffffffffffffffffffff
```

owner: 0x0000000000000000000000000000000000000000*address*

tokenBalanceForAddress: mapping(address => mapping(uint8 => uint256))

balanceEthForAddress: mapping(address => uint256)

tokens: mapping(uint8 => struct Exchange.Token)

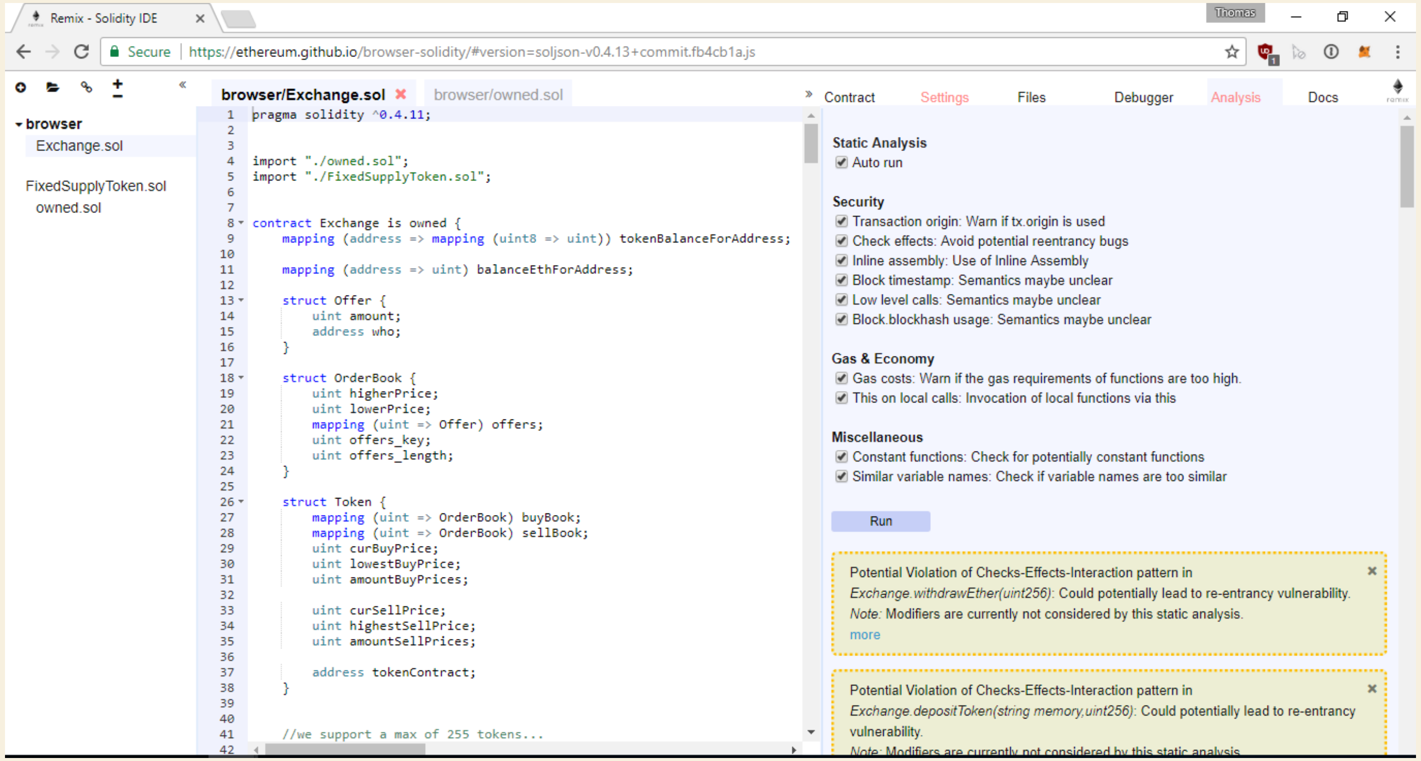
symbolName: mapping(uint8 => string)

ANALYSIS

- Warnings when „unsafe“ code is used
  - Tx.origin
  - Or address.send(…)
  - …
- Warnings about gas costs
  - When they are too high

- Check if functions could be constant
- Check for similar variable names

ANALYSIS



```
pragma solidity ^0.4.11;


import "./owned.sol";
import "./FixedSupplyToken.sol";


contract Exchange is owned {
    mapping (address => mapping (uint8 => uint)) tokenBalanceForAddress;

    mapping (address => uint) balanceEthForAddress;

    struct Offer {
        uint amount;
        address who;
    }

    struct OrderBook {
        uint higherPrice;
        uint lowerPrice;
        mapping (uint => Offer) offers;
        uint offers_key;
        uint offers_length;
    }

    struct Token {
        mapping (uint => OrderBook) buyBook;
        mapping (uint => OrderBook) sellBook;
        uint curBuyPrice;
        uint lowestBuyPrice;
        uint amountBuyPrices;

        uint curSellPrice;
        uint highestSellPrice;
        uint amountSellPrices;

        address tokenContract;
    }


    //we support a max of 255 tokens...
```

Static Analysis

☑ Auto run

Security

☑ Transaction origin: Warn if tx.origin is used
☑ Check effects: Avoid potential reentrancy bugs
☑ Inline assembly: Use of Inline Assembly
☑ Block timestamp: Semantics maybe unclear
☑ Low level calls: Semantics maybe unclear
☑ Block.blockhash usage: Semantics maybe unclear

Gas & Economy

☑ Gas costs: Warn if the gas requirements of functions are too high.
☑ This on local calls: Invocation of local functions via this

Miscellaneous

☑ Constant functions: Check for potentially constant functions
☑ Similar variable names: Check if variable names are too similar

Run

Potential Violation of Checks-Effects-Interaction pattern in *Exchange.withdrawEther(uint256)*: Could potentially lead to re-entrancy vulnerability. *Note:* Modifiers are currently not considered by this static analysis.
more

Potential Violation of Checks-Effects-Interaction pattern in *Exchange.depositToken(string memory,uint256)*: Could potentially lead to re-entrancy vulnerability.
Note: Modifiers are currently not considered by this static analysis.

WHAT YOU LEARNED HERE

- Remix is a web IDE for smart contract development
- Has a convenient step-by-step debugger
- Integrated simulated ethereum protocol simulation
- Can also speak to web3 if injected or a HTTP-RPC