CMSC 398F
Week #9
Intro to Smart Contracts and Solidity

•••

Announcements

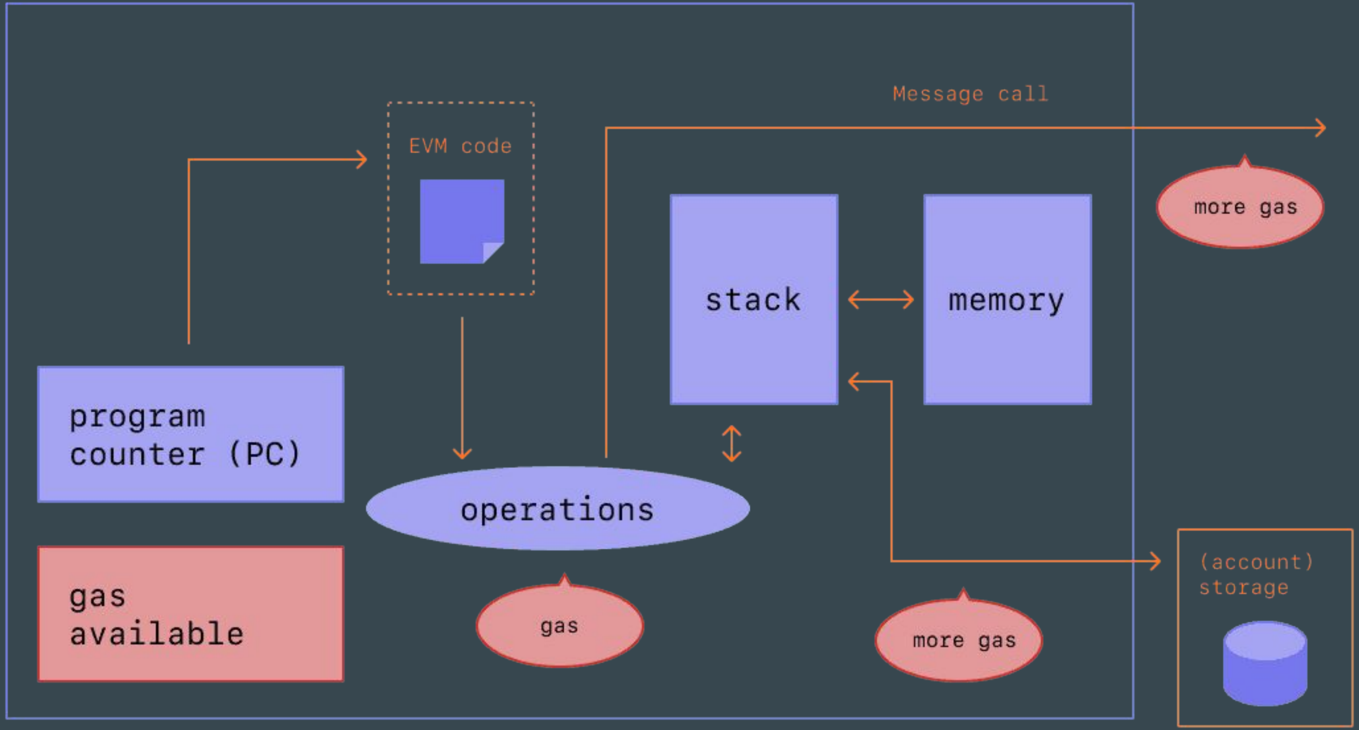
- Join the class Piazza!
  - piazza.com/umd/fall2022/cmsc398f
- Exam 1 Grades out
  - If we misgraded something, please make a **private Piazza post**
  - Everyone gets full credit for question 6

Recap

- Altcoins and different types of altcoins
- Ethereum
- Proof-of-Stake vs Proof-of-Work
- Regulations of Cryptocurrency around the world

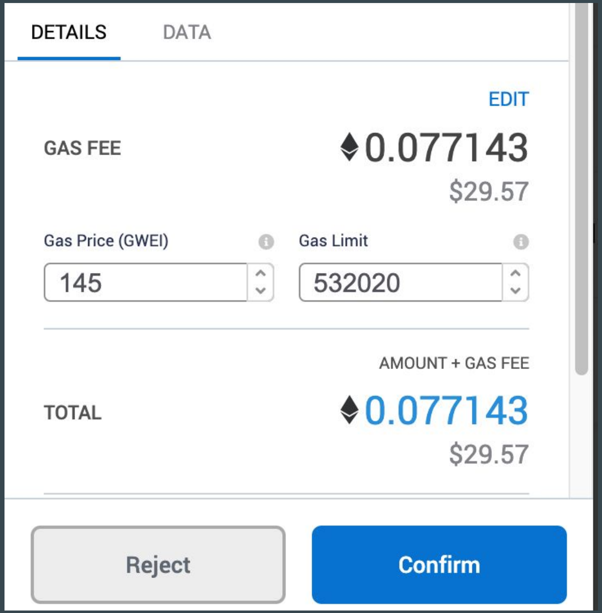
Ethereum Virtual Machine (EVM)

- Instead of a distributed ledger, Ethereum is a distributed state machine
- Ethereum's state is a large data structure which holds not only all accounts and balances, but a machine state, which can change from block to block according to a pre-defined set of rules, and which can execute arbitrary machine code.
- The EVM behaves as a mathematical function would: Given an input, it produces a deterministic output.


Message call
EVM code
more gas
stack
memory
program counter (PC)
operations
gas available
gas
more gas
(account) storage

Gas and Fees

- Gas refers to the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network.
- When you pay gas to submit a transaction, you are paying for the computational energy needed to power the validation of that transaction on Ethereum.
- A key component of the Ethereum gas system is the Ethereum gas limit. In the context of transactions, the gas limit is the maximum amount of gas units you are willing to spend on a transaction



GAS FEE ♦0.077143 $29.57

Gas Price (GWEI): 145

Gas Limit: 532020

AMOUNT + GAS FEE

TOTAL ♦0.077143 $29.57

Introduction to Smart Contracts

- A "smart contract" is simply a program that runs on the Ethereum blockchain.
- Smart contracts digitize agreements by turning the terms of an agreement into computer code that automatically executes when the contract terms are met.
- Think about contracts in the real world
    - Kickstarter: fundraising platform. Product teams can create a project, set a funding goal, and start collecting money from others.
    - Both project teams and supporters trust kickstarter to manage their funds.
- We can implement this with a smart contract
- We can program a smart contract so that it holds all the funds till a goal is reached. If a program meets its goals, the project owners get the money, if not the money goes back to the supporters.
- Releasing funds is just one example, smart contracts can even do things like send notifications.

Why are Smart Contracts Important

- Speed, efficiency, and accuracy
    - Once a condition is met, the contract is executed swiftly. No paperwork to process and no time spent reconciling errors that often result from manually filling in documents.
- Trust and transparency
    - There is no third party involved, and contracts on the blockchain are immutable.

Applications of Smart Contracts in real-world

- The applications of smart contracts are endless
- Banks
  - Issue loans or automatic payments
- Insurance companies
  - Use it to process claims
- Postal companies
  - Process payment upon delivery
- Increasing trust in retailer supplier relationships
  - The Home Depot use smart contracts to resolve disputes with vendors
- IBM Blockchain
  - Making global trade faster and more efficient

How to develop Smart Contracts - Solidity

- Solidity is a type of Object-oriented programming language that is developed specifically for smart contracts on the Ethereum blockchain
    - Used to implement smart contract features on various blockchain platforms
- Influenced by C++, Javascript, and Python and is statically-typed
- Solidity uses the EVM to function properly
    - EVM offers a runtime environment for smart contracts to execute
    - Solidity is compiled to bytecode that is executable on the EVM

Different ways of setting up a Solidity Compiler

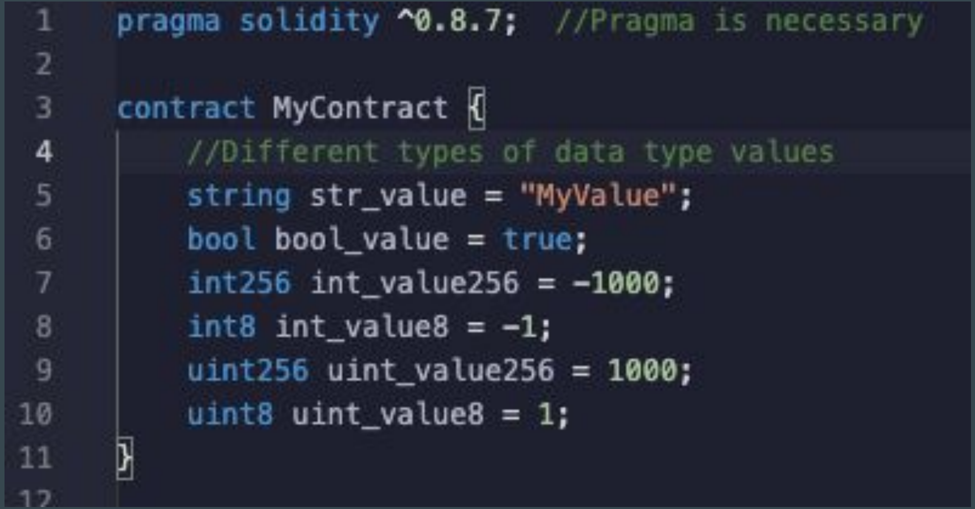
- Remix
    - an application that provides plugins and a development environment for smart contracts. Users can use this application online without installing any software for the environment
- Node.js/npm
    - you will npm install a solidity compiler called solc-js. However, solc-js offer limited functionalities for accessing the compiler
- Docker Image
    - Docker image offers simple steps in setting up the environment. Docker images offer a template to build a container for running the Solidity compiler
- Binary Packages
    - Binary packages are archive files that will have all directories and files for installing the Solidity compiler on your device

Solidity Syntax

- Pragma
  - In Solidity, a pragma language will specify how the compiler will process any type of input
  - Typically, the first line of code in Solidity based smart contracts contains the pragma to specify the Solidity version
    - pragma solidity ^0.8.7;
- Contract
  - A Solidity contract is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum Blockchain
- File Importing
  - Solidity offers similar support for file import systems like JavaScript.
    - import "filename";

Data types in Solidity

- 8 bits to 256 bits Unsigned integer
- 8 bits to 256 bits Signed integer
- Boolean
- String



```
pragma solidity ^0.8.7;  //Pragma is necessary

contract MyContract {
    //Different types of data type values
    string str_value = "MyValue";
    bool bool_value = true;
    int256 int_value256 = -1000;
    int8 int_value8 = -1;
    uint256 uint_value256 = 1000;
    uint8 uint_value8 = 1;
}
```

Variables

- State Variable
  - Users can locate the state variables within the contract storage where values are permanently stored
- Local Variable
  - Users can find the value of any local variables within the defining function. This value is not permanently stored
- Global Variables
  - Global variables help in fetching data or information from the blockchain platform and any associated transaction processes
  - Common Global variables that might be used:
    - msg.sender
    - msg.value
    - this
    - block.timestamp
    - block.difficulty

Visibility Modifiers

- Visibility modifiers define the visibility of state variables or functions
- There are 4 modifiers that Solidity has:
    - External
    - Public
    - Internal
    - Private
- We will talk about these in more depth next week!

https://cryptozombies.io/

- This is a free bootcamp for you guys to learn Solidity
- We will be teaching some important things as we continue, but not everything
  - But we expect you guys to do to assigned CryptoZombies.io each week
  - The reason for this is for you guys to get your own hands-on experience with Solidity
- Quizzes each week on Solidity from the assigned CryptoZombies.io lesson
- Next Week: Finish Lesson 1: “Making the Zombie Factory”