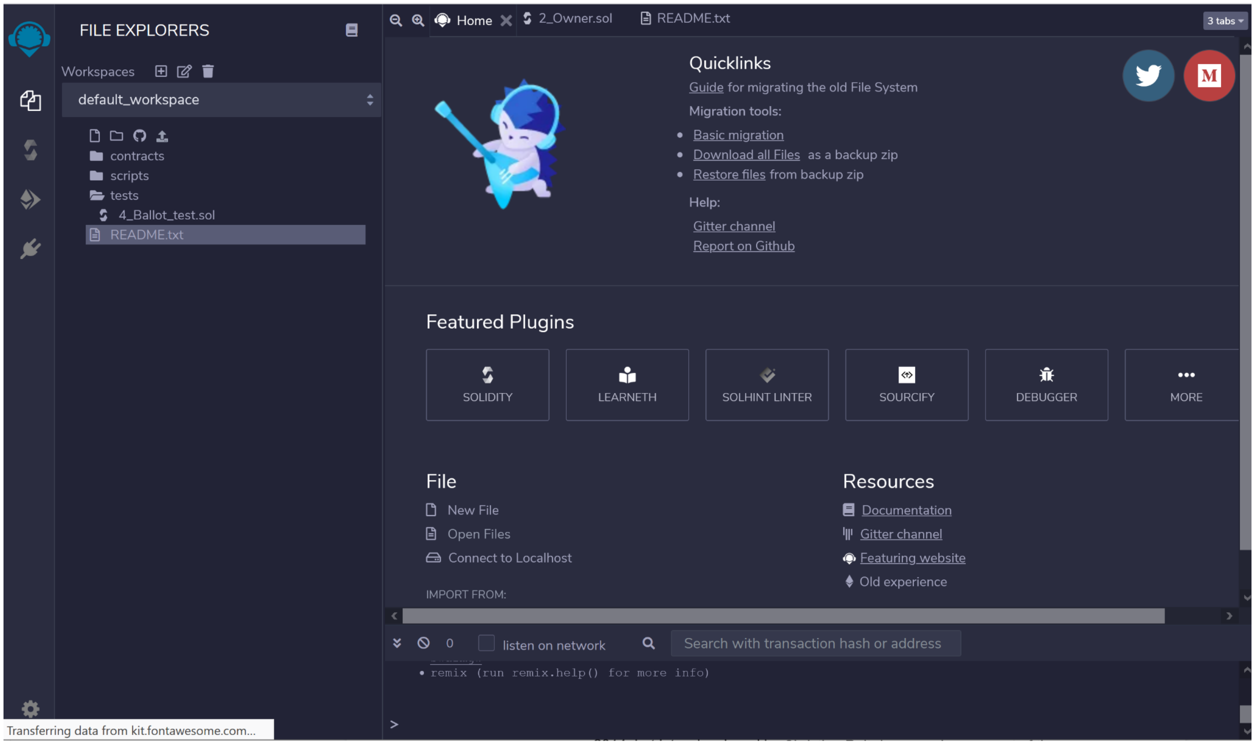
Lexie Rista, Archibald Latham, Chiedozie Anazia
CSC 421- Design and Organizing Programming Languages

https://american.zoom.us/j/91910115418

Platform

- https://remix.ethereum.org/
    - This is a great source for initially learning the language.
    - It is accessible with a web browser so all of us are able to use this environment now. This is probably where we will run our code. If we run into a problem, we found another link located below.



Guide for migrating the old File System

Download all Files as a backup zip

Restore files from backup zip

```
• remix (run remix.help() for more info)
>
```

Transferring data from kit.fontawesome.com...

Figure 1. Remix.ethereum.org home page

WE COULD EVEN HAVE ANNOTATIONS (LIKE RED CIRCLES) HIGHLIGHTING THE IMPORTANT FEATURES TO BE USED

- https://github.com/ethereum/remix-live/tree/gh-pages
    - If we decide that we want it downloaded onto our machines, this is where we can get it. However, I believe most of our use will be with the first link.

Background

- Who designed/designs the language?
    - Initially proposed by Gavin Wood the co-founder of Ethereum(ETH) in 2014, but later developed by Christian Reitwiessner who was part of the Ethereum project's Solidity team(Wiki)
    - Etheruem was used as a direct competitor to Bitcoin
- Is there an official standard?
    - Solidity is the most popular smart contract language with few competitors
    - Java, Javascript, Python, Go are also used in the development of ETH
- Is there one implementation or multiple?
    - There are multiple real life implementations of Solidity. All ERC-20 tokens use this language to interact with the ethereum blockchain.

** ADD FIGURE: EXAMPLE OF WHAT A BLOCKCHAIN LOOKS LIKE (SPECIFICALLY IN ETHERUEM)

- How are revisions to the language made, and who decides?
    - There is a Solidity forum (https://forum.soliditylang.org/) where users are able to propose and discuss language changes and new features. After proposed features become more concrete, they will be discussed on the Solidity GitHub Repository. It seems like anyone can join in on the language design conference calls (invitations to these meetings are posted in the forum). The team and contributors have the final say in what will be implemented in the language.
- What sort of community is there around it, and who does it target?
    - Large community within the realm of cryptocurrency as the language is used to write decentralized applications such as Uniswap and Aave

Language Syntax and Semantics

- What does the language look like?
    - The language is very similar to java and has influences from python as well as C++
    - It is statically typed(i.e Java, C, C++), but has syntax similar to JavaScript.
    - It's also type-safe unlike Javascript.
    -

** ADD FIGURE: COMPARISON OF SOLIDITY, JAVA, PYTHON (ESSENTIALLY BE À CODE THAT DOES THE SAME THING, BUT THIS IS JUST TO SHOW SIMILARITIES IN SYNTAX OR STYLE)

- How does it work?

        - The language works like an other language except its output is printed to the ethereum blockchain and can be interacted with by anyone
        - The Ethereum Virtual Machine which behaves as a giant conducts transactions automatically which is nothing more than transporting a series of opcode or instructions to different nodes within the network.
        - Mining takes place when each group of transactions sent by the users, which are the smart contracts, are validated by the nodes and then are appended to the blockchain.
        -

** ADD FIGURE: SHOW WHAT THE OUTPUT INTO ETHEREUM BLOCKCHAIN LOOKS LIKE

Analysis of language

- A deeper dive into at least a few specific features of the language (with example code)
    - What makes it unique?
        - The language is used to write smart contracts on the Ethereum blockchain.
        - Ethereum, like Bitcoin, is a blockchain platform that handles the trading of cryptocurrencies (which in this case is Ether). The foundation of Ethereum's network is Solidity. So Ethereum will cease to exist without Solidity.
    - Highlight what features make it different from a general-purpose language
        - The language implements features that enable it to interact with the decentralized ledger.
        - Solidity is able to create contracts for voting, crowdfunding, blind auctions, and multi-signature wallets.
        - Ethereum Virtual Machine (EVM) hosts Ethereum for the smart contracts' runtime environment. This leads to smart contracts being isolated from outside networks, filesystems, or processes.
    - What features do we plan on investigating further?
        - Writing a contract that will interact with an Ethereum test network and possibly create our own test token.
    - Security
        - The prominence of the Ethereum blockchain have led to several security issues. In order mitigate damage or to defend against possible attacks, Solidity has better has implemented better security design patterns to assure safe transport of instructions.(Add examples)
            - Design Patterns
                1. Checks and Effects Interactions - A design pattern preventing reentracy attacks by making sure the address call is the last step of code in order to avoid any type of external interaction. The unwanted interaction between functions is usually the root cause of this, but limiting the

code execution by minimizing the transfer of funds(ether) protects the contract from malicious operations.

2. Emergency Stop/Circuit Breakers - Simple way to stop contract code execution once a malicious agent is detected. For example, if the a bug is detected during the transfer of ether, then all the operations are halted in order to protect against a loss of any remaining assets. The emergency stop disables the function within the contract.

- SafeMath
    - require ()
- Unique address (data type)

Discussion

- What was our experience with this language?
- What did our code produce?

** ADD FIGURE: WHAT DID WE CREATE WITH OUR CODE

- With this product, what can it be used for? What are the next steps?

** ADD FIGURE: WHAT CAN WE DO WITH WHAT WE CREATED

- Why is this language important? (for the future)
- Security
    - Approve and transfer
- Privacy

Resources:

https://docs.soliditylang.org/en/v0.8.3/

https://docs.soliditylang.org/en/v0.8.3/structure-of-a-contract.html

https://cryptozombies.io/

https://remix.ethereum.org/

https://github.com/vahiwe/Building-your-own-ECR20-Token

https://www.bitdegree.org/learn/solidity-require

Articles:
https://arxiv.org/ftp/arxiv/papers/1803/1803.09885.pdf
https://arxiv.org/pdf/1907.02952.pdf
http://eprints-dev5.cs.univie.ac.at/5433/7/sanerws18iwbosemain-id1-p-380f58e-35576-preprint.pdf