Marvin Chan

Final Project Progress Report

Overview:

The invention of the blockchain has allowed for a new computing paradigm - a programmable computer that can be used by anyone but is not owned by anyone - and code deployed on the blockchain is called a smart contract. Solidity is by far the most popular programming language for smart contract development with 8,000+ public repositories on GitHub. Solidity provides developers a lot of flexibility, but this flexibility has also resulted in hacks in the hundreds of millions of dollars. As a result, developers were looking to develop a new language that limits security flaws and came up with Vyper. Vyper is much newer than Solidity, and the first stable version of Vyper was released in 2020. Hence, there are only 84 public Vyper repositories on GitHub.

For my final project, I plan to quantify how long and how difficult it is to learn to develop a smart contract in Vyper by reimplementing Uniswap V2 smart contract. Uniswap V2 smart contract, which is written in Solidity, facilitates swapping between cryptocurrencies and it is the 3rd most widely used smart contract on Ethereum. I chose to reimplement an existing Solidity smart contract because the features are well defined making it easier to test correctness. I also chose Uniswap V2 instead of Uniswap V3 because V3 has additional features for trading. Reimplementing these additional features would mainly just be writing additional application logic, but does not add value to the goal of quantifying how long it takes to learn Vyper. To quantify the process, I will keep track of the amount of time spent reading Uniswap V2, Solidity documentation, Vyper documentation, and the Google searches I made to learn concepts, syntax, etc.

Value to User Community:

The target audience for this final project is experienced developers who want to learn smart contract development. I believe my journey of learning Vyper would be helpful to developers interested in learning smart contract development because my background matches the background of many people trying to get into the blockchain industry. I have worked a couple of years as a Data Engineer and am about to complete an MSCS. Hence, I would generalize myself as an experienced developer. I have also been a user of the blockchain since 2021, a year when Metamask, a wallet company, saw monthly active users grow from 500,000 to 10,000,000. As a result, my profile fits many of the people that are thinking of going into smart contract development. This project would provide more information for new smart contract developers to help them choose whether to learn Solidity or Vyper.

Research Questions:

1. How difficult it is to learn Vyper when there are limited examples on GitHub and answers on Stack Overflow?
2. Is it better to learn Solidity first before learning Vyper?
3. How long does it take to learn Vyper without learning Solidity?

Demo:
For the demo, I plan to provide a quick overview of blockchains, smart contracts, Solidity, Vyper, and Uniswap. I will then go over cases I have come across where limited documentation and examples made it more difficult to develop in Vyper.

Other Information:
How code, documentation and other software artifacts are delivered:
https://github.com/marv-chan/COMS_6156_Final_Project

References

Uniswap v2:

- Smart Contract Code: https://github.com/Uniswap/v2-core
- Documentation: https://docs.uniswap.org/

Curve (example Vyper project I can reference):

- https://github.com/curvefi/curve-contract

Solidity: https://docs.soliditylang.org/en/v0.8.13/
Vyper: https://vyper.readthedocs.io/en/stable/index.html