Introducing UniStark: Uniswap, only Warp’ed to StarkNet

Jorik Schellekens

Follow

Nethermind.eth

--

1

Listen

Share

By Jorik Schellekens

Thanks to

[Guy McComb

](https://twitter.com/0xGai_),

[Greg Vardy,

](https://twitter.com/0xGreg_)

[Swapnil Raj

](https://twitter.com/swp0x0),

[Carmen Irene Cabrera Rodriguez

](https://twitter.com/cicr_99),

[Rodrigo Pino

](https://twitter.com/rodro_pino), [Anneli Nowa

k](https://twitter.com/ojneloy), and

[Carla Segvic

](https://twitter.com/carlaaseg)for helpful comments and revisions.

We are excited to announce that the next big milestone in the Warp project has been reached: we have successfully transpiled and compiled Uniswap v3! What’s more, is that we’re finishing a hardhat plugin that allows you to run all your Solidity hardhat tests on the transpiled Cairo:

In the video above, we’re using the original

SqrtPriceMath.spec.ts

file from the Uniswap v3 hardhat test suite. The hardhat Warp plugin automatically runs the tests on the transpiled Cairo. The plugin is still under development, but soon we will have Uniswap’s entire test suite running against a Uniswap implementation deployed on StarkNet.

From day one, Warp was designed to allow users to transpile existing Solidity codebases to Cairo and deploy them on StarkNet. With the recent advances on StarkNet, namely contract creation from other contracts, Warp is finally in a position to go after one of Solidity’s most complex projects: Uniswap v3.

Don’t take our word for it; go to[our Uniswap fork](), install the latest[Warp]() and take it for a spin!

Warp successfully transpiles and deploys every Solidity file in the[Uniswap v3 repo]() with only minor changes to the source.

Much work has gone into correctly compiling a project of this size, but the effort pays off-net. With Warp rapidly maturing, the barrier to entry for large and small projects alike to test out the capabilities of StarkNet is lowering.

Warp is not about to stop with Uniswap! We will continue working hard on features and repeat this experiment with a few other protocols, bringing new protocols to StarkNet at warp speed. If you are interested in Warping your project to StarkNet, please [reach out]()!

Attempting this transpilation has ironed out many kinks in Warp's usage and helped us to prioritize new features. It's also motivated us to integrate Warp with Hardhat — more on this later.

As a result of this effort, Warp now also supports the following:

- abi.encode

, abi.encodePacked

- Caution! Addresses are packed in 32 bytes because StarkNet addresses are 251 bits
- abi.decode
- For now, only value types are supported