When we release this Local Dev Testnet software, we'll want devs to be able to debug their Noir Contracts (and their entire dapps) easily. Here's an open list of questions.

If anyone has any strong experience or opinions on these subjects, please do make suggestions in this thread! I'd encourage you to also make a separate post on this forum, if you have a nice proposal (even if you haven't figured out all the details yet), so that it can be debated and improved more easily – this will also help reduce clutter in this thread!

Noir currently enables users to write unit tests for individual circuits (functions).

But what's the best way of enabling users to test calls between functions (both within the same contract or between two different contracts)?

And how can we give as much information back to the user as possible?

- stack traces
- contract addresses, function names, args, return values, emitted events, etc.
- contract addresses, function names, args, return values, emitted events, etc.
- events
- console.log/debug functionality within a Noir Contract?
- error messages

I haven't played with Foundry at all, but I know it's got some nice 'native Solidity' testing features vs hardhat / truffle etc. To what extent can we enable 'native Noir' testing for users, versus them writing tests in typescript?

What are the tools and features that devs expect, when writing and testing Solidity contracts, these days?

Do we need advanced features like pausing the blockchain, rewinding it, etc?

Cheers!