Ethereum Audio

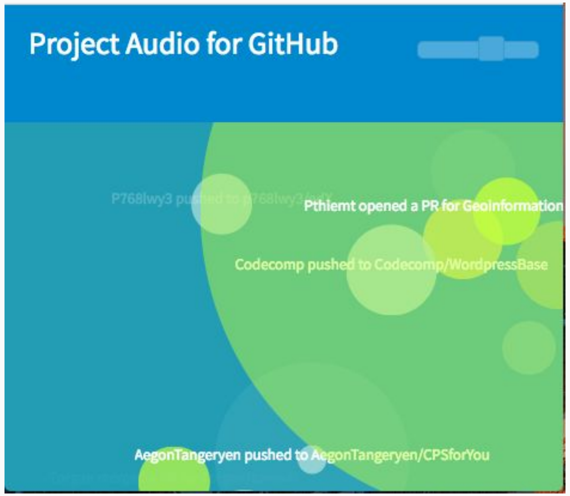
Roey Shamir & Itay Radotzki

The need

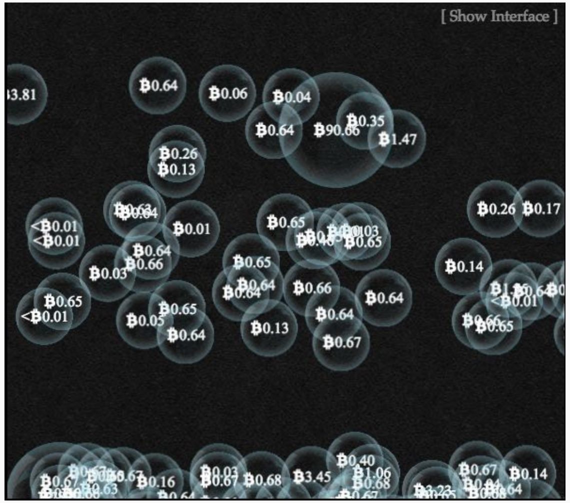
- The Ethereum network is huge - lots of transactions and blocks.
- It is hard to **grasp** the magnitude of the transactions made in Ethereum with traditional tools.
- Right now this is impossible to listen to the interesting data from this network.
- Everyone can use Ethereum Audio in order to get a feeling of the usage in Ethereum based apps.
- Miners can use Ethereum Audio in order to set their fees.

Market Research

https://github.audio/


Project Audio for GitHub
P768lwy3 pushed to p768lwy3
Pthiemt opened a PR for Geoinformation
Codecomp pushed to Codecomp/WordpressBase
AegonTangeryen pushed to AegonTangeryen/CPSforYou

http://www.bitlisten.com/



Our Product

- A web application that tracks events in the Ethereum network and converts them to music notes based on certain parameters.
- Everyone, at any age, who is interested in Blockchain technologies will like Ethereum Audio.
- While using Ethereum Audio, one can analyse the Ethereum network, have a better understanding of the amount of blocks and transactions right at the moment.

Our Solution

When listening to the Ethereum network using Ethereum Audio, it is extremely easy to get a sense of the current usage in the Ethereum platform (Real-time).

- Dedicated Ethereum node which mines and broadcasts transactions.
- Sound mapping
    - Transactions - the pitch and timing determined by the transaction’s amount.
    - Blocks - the pitch is determined by the number of txs in the block, and the timing by the time that was taken to mine the block.

Competitive Analysis

Stock sonification is everywhere. However blockchain is a relatively new field with much more parameters one can sonify.

Only with Ethereum Audio you can:

- Listen to the **Ethereum network.**
- Listen to the **time** that was taken to **mine a block.**
- Listen to the **number of transactions in a block.**

And all in real-time!

Architecture

(1) Transactions Generator
Generate random transactions and send them to the Ethereum network.

(2) Node
A local Ethereum node that mines blocks.

(3) Server
Subscribe to new blocks and new transactions, broadcast them via WebSockets.

(4) Client
Subscribe to server events and play a sound for each new event (block/transactions).

Sprints

Sprint I
Build the infrastructure of the project, check how to run a local Ethereum node, create a server with WebSockets broadcaster and a client with WebSockets listener.

Sprint II
Server should subscribe to events from the local Ethereum node and broadcast them to the client. The client should play sound according to the data that was received. In addition build a tool that will generate random transactions and send them to the local node.

Sprint III
Sound mapping enhancement, build UI that represents the sound in a visual way. Fine tuning.