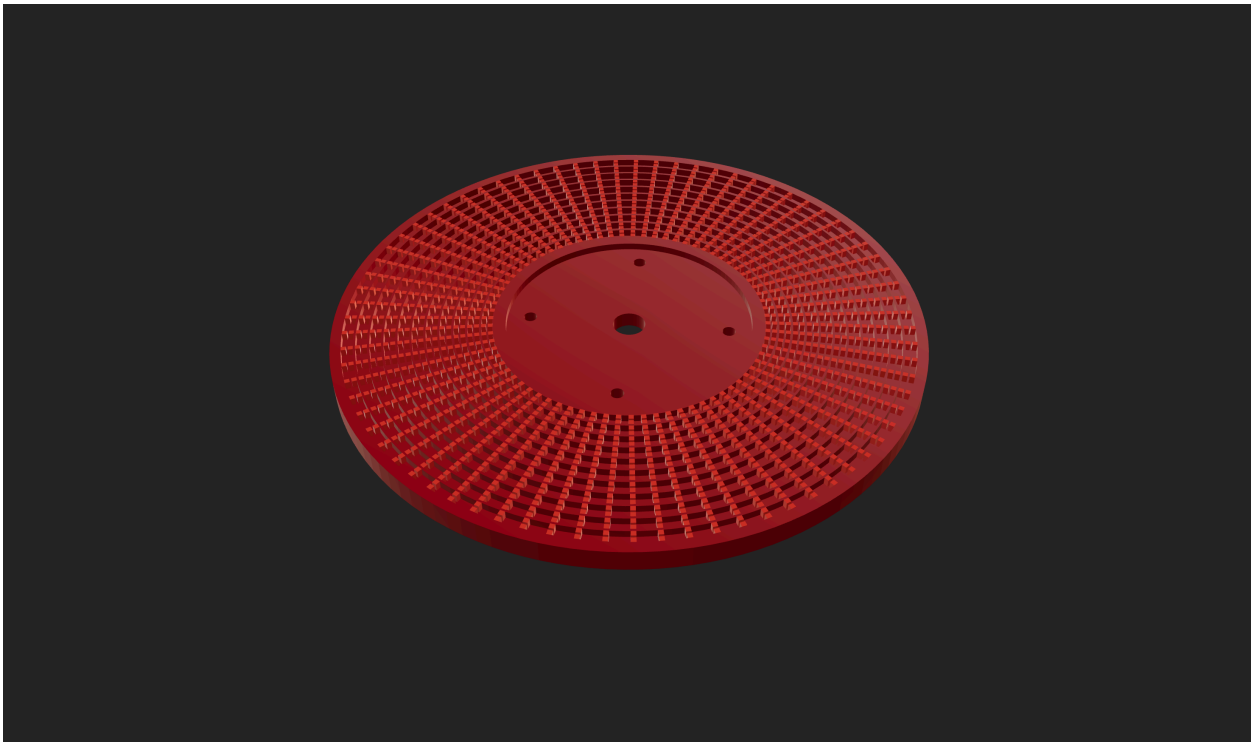
PROTOTYPE

Tentatively: Fisher-Juke

FisherJuke as it is tentatively titled, differs little from the proposal last submitted. In essence, the project involves the programmatic creation of Fisher-Price diskettes as STL files. These files are then able to be immediately 3-D printed as they are scaled appropriately and offer correct dimensional fidelity with regards to whatever arbitrary data a user may want to insert. The primary interface for the creation of the is a web interface using *Threejs* to represent the disk as it's being edited. The data component of this project, where a user may browse other peoples creations from an online library is underway, but it's not yet built. However, the main logic for the creation of the diskettes has been implemented and is live at the link below.



Above: Initial disk prototype with every one of the possible ridge placements filled.

Tech stack & Progress

Still to be built, is the broader UI for the inputting of notes and their playback, as well as the data formatter that compresses arbitrary data into the highly quantized ~3000 bit format of the diskette. Additionally, following feedback, a

new stretch goal of implementing an audio decoder that translates sound from the Ficher-Price toy into the portable STL format will be attempted.

The project is underway with a tech stack consisting of Svelte, *Threlte (for threejs interop)* and standard CSS styling. Database concerns will very likely be handled by a Mongo Atlas instance. The format for data storage is so far an array, 75 items long, comprised of text strings resembling binary, representing either the presence or absence of a ridge at a particular position in the disc. Improvements could be made to store the string as a hexadecimal number to then convert to binary, to then convert to a string / swizzle the one and zeroes into place.

Notably, during development of this project, I discovered and fixed an issue with the three-stdlib STL exporter.

The live demo will generate an STL file with random data at the time of writing. It may be used after installation following source code *here*.

Thanks!