Mel Aise (they/them) // email: mochilemon@proton.me, personal site: pyon.moe, code: github.com/yurapyon

Employment experience:

Monterra, Oct 2023 - Jan 2026 // https://www.monterra.ai/, 2D Editor Demo

Monterra is a VC-funded startup building a web-based visual design tool for EV charger installers. The app also performs cost analysis, automates proposal/quote generation, and verifies code-compliance.

I was the second hire and helped move the product from prototype to a production-ready app. Led a project to rebuild the 2D graphics editor from the ground up, greatly improving its maintainability and performance. Updated frontend React code to match UI/X designs from a designer, drawing on my experience with CAD and digital art software to fill in any gaps. Designed developer-friendly data models for implementing electrical analysis. Planned projects for myself and others, including large DB changes and coordination between the web/mobile teams. Advocated for good practices in a more feature-driven environment.

Technologies: Typescript, React, Next.js, Tailwind, Supabase, PropelAuth, Webhooks, Mapbox, Pixi.js

Ping, Nov 2020 - Jan 2023 // https://ping.gg/

Ping is a Y Combinator backed startup building a low-latency video conferencing app for live-streamers.

I was the first non-founder hire, and this was also my first job in the industry. Designed and implemented a UI for managing video call participants and visual customization of the feed. Planned and built AWS infrastructure to record, transcode and store video streams. Helped with R&D on an 'anonymous questions' app. Also, sent sales emails and did market research.

Technologies: Typescript, React, Next.js, Tailwind, tRPC, Prisma, NextAuth, Ably, Agora, AWS

Before the software industry, Before 2020

I've worked in various small family-owned restaurants. Have experience working under pressure, reprioritizing on-the-fly, and collaborating closely with coworkers and managers.

Creative background:

Self-taught programmer, 2015 - Ongoing

I didn't have money for college after high school so I taught myself programming from resources online. Started around 2015 using Unity and C# to make a game. I wanted to go deeper, so I wrote a graphics engine for making video art, using C, Scheme and OpenGL. Then, I learned functional programming and algorithms. Eventually returned to low-level code, picked up Zig and Forth, and continue to work on game dev projects.

Technologies: C, Zig, Rust; Scheme; Lua; Unity, Godot; Linux; Functional programming, Algorithms

Self-taught artist/musician, 2012 - Ongoing

I love visual art and music! I have spent a lot of time producing music and working on 3D art. My experience with audio and visual design tools has carried over considerably into designing software UI/X.

Technologies: Ableton, Renoise; Blender, Clip Studio Paint; Photoshop, Gimp, Pencil and paper

Selected personal project:

mini // https://github.com/yurapyon/mini, Web Demo

mini is a specification for a self-hosting 16bit Forth system and a Zig bytecode executor + virtual PC.

I started this project because I like Forth a lot! Practicality and clever elegance are core philosophies behind Forth, and it pushes me to hone my ability to "get more done with less code". The Forth community is pretty fragmented, so I spent a lot of time studying open source implementations as inspiration for my own.

This project is also an exploration into old-school computers. The virtual PC is modeled after the Japanese PC-98 and I find it fun to work under hardware limitations. I recently got the library working on the web using WebAssembly and Solid.JS. In the future I want to get it running on an Arduino UNO and build out an easy-to-learn API for writing games.

Technologies: Forth, Zig, GLFW, OpenGL, WebAssembly, Portaudio, Solid.JS