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

Employment experience:

Monterra // https://www.monterra.ai/, October 2023 - Current

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

I was the second hire and helped move the product from prototype to a production-ready app. This required rebuilding the 2D graphics editor from the ground up; redesigning data models and restructuring the DB; updating frontend code to match UI/X designs from a designer; building developer-friendly frameworks for implementing features to analyze electrical designs; improving code reliability, maintainability and organization; and more.

I drew from my experience with CAD, EDA and digital art software to help inform UI/X decisions. I also relied on linear algebra concepts I'd learned from game development when reworking the design editor.

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

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

Ping is a Y Combinator backed startup making a video conferencing app for live-streamers. The app offers 1080p, low latency video; built-in recording; and built-in collaboration management tools.

I was an early hire, and this was also my first job in the industry. I designed and implemented a fullstack feature for visual customization of video calls. I also worked on AWS infrastructure to record, transcode and store video streams. When not doing that, I would send sales emails and do market research.

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

Before the software industry // Before 2020

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

Creative background:

Self-taught programmer // github.com/yurapyon, 2015 - Ongoing

I didn't have money for college after high school, but I managed to teach myself programming from resources online. I started around 2015 using Unity and C# to try and make a game. This eventually turned into wanting to write my own game engine, so I spent time learning C and OpenGL. After a couple years doing that I learned functional programming, algorithms, and Scheme. Eventually I went back to focusing on low-level code, picked up Zig and Forth, and have continued working on game dev and microcontroller projects.

Technologies: Zig, Forth, Scheme, C, Lua, Unity/Godot, Functional programming, Algorithms

Self-taught artist/musician // pyon.moe, 2012 - Ongoing

I love visual art and music! After high school I spent a lot of time producing music and working on 3D art. At times I've tried to bring together art and programming to make videos or games. These days I'm studying drawing and learning character design.

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

Selected personal project:

mini // https://github.com/yurapyon/mini

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 writing Forth, and it really pushes me to hone my ability to "get more done with less code". Working with unstructured/untyped data challenges you to rethink fundamental concepts of programming.

The Forth community is pretty fragmented, so I spent a lot of time studying different Forth implementations as inspiration for my own. I started with a Zig-based interpreter, but over time I was able to make it self-hosted. Due to this most of the runtime is written in Forth, and all the system needs to run is something to execute bytecode. In the future I'd like to write an executor for the Arduino UNO and build a modern "retro" computer on top of it.

This project is also an exploration into working with old-school computers. The virtual PC is modeled after the Japanese PC-98 and I find it fun to work under hardware limitations, such as fixed screen resolution and limited system memory.

Technologies: Forth, Zig, GLFW, OpenGL