Elliot Miller

734-223-7401 | Website | elliotgrantmiller@gmail.com | linkedin.com/in/elliotgmiller | github.com/elliotmiller18

Education

University of Michigan Ann Arbor, MI

Bachelor of Computer Science *Expected Dec 2026*

- Relevant Coursework: Formal Verification of Systems Software, Operating Systems, Compilers, Web Systems, Computer Organization, Data Structures and Algorithms, Computer Theory, Game Development, Programming Languages

Experience

Software Engineering Intern (Rust) May 2026 – Aug 2026

Apple *Seattle, WA*

- Incoming Software Engineering intern working on distributed block storage.

Systems Engineering Intern (Rust) Jan 2026 – May 2026

N1 (Founder's Fund) *New York, NY*

- Contributing high-impact, production code to a rapidly evolving, highly performant CLOB matching engine handling hundreds of millions of dollars in volume.

Software Engineering Intern (Typescript) May 2025 – August 2025

Wise Pelican *Phoenix, AZ*

- Wrote a from-scratch fast JPEG metadata decoder, cutting off over 95% of runtime from the previous implementation by reducing data needed to process per JPEG from megabytes to a single kilobyte
- Decreased average runtime of PDF rendering engine from 10s to 2s by moving image generation to an asynchronous model, allowing smaller requests to be processed, generated, and responded to in the time that larger PDFs were rendering

projects

GameBoy Emulator (C++) Github

- Simulated all GameBoy-specific hardware components/systems, including internal clock circuits, interrupts, and proprietary graphics
- Wrote a fully accurate Z80 Sharp CPU simulator/machine code interpreter to run ROMs, emulating hardware bugs involving finicky instruction timing and interrupt handling
- Achieved consistent a consistent 300 frames per second using a custom SDL-integrating renderer on an M1 Macbook with integrated graphics

ChAsm (C, ARM Assembly)

- Wrote a fully from-scratch (no standard lib) graphical chess implementation in ARM assembly supporting real-time play along with a C/SDL event bridge/renderer.
- Optimized with SIMD instructions and by keeping total memory usage under a single L1 cache line.
- Used no external libraries by implementing all core game and rendering logic from scratch in assembly

Snake Compiler (Rust, x86)

- Wrote an x86 emitting, optimizing compiler for a Python/OCaml hybrid language called Snake

Multithreading Library (C++)

- Created an efficient implementation of Mutexes and CVs similar to that of the C++ STL in a toy kernel that leveraged a simulated underlying multi-core CPU for concurrent execution

Miscellaneous

- Was ranked #3 on the Stack Overflow global leaderboards (see my website for proof)

Technical Skills

Languages: C++, C, Rust, x86_64, ARM, Dafny, Python, C#
Libraries/Frameworks: SDL.h, AWS CDK/SDK, Unity
Developer Tools:Git, AWS, AWS Lambdas, Docker, MacOS, Linux, NVim