

# Elijah Baraw

[ebaraw@andrew.cmu.edu](mailto:ebaraw@andrew.cmu.edu) | (203) 731-9535 | Pittsburgh, PA | [github.com/elijah-bae-raw](https://github.com/elijah-bae-raw)

---

## EDUCATION

**Carnegie Mellon University, School of Computer Science**

**Aug 2021 – May 2025**

*Bachelor of Science in Computer Science. Concentration in Computer Systems*

GPA: 3.94. Relevant courses: Machine Learning, Cloud Computing, Distributed Systems, Data Structures and Algorithms, Functional Programming, Systems, Parallel Algorithms, Linear Algebra, Differential Equations, IDL

## TECHNICAL SKILLS

**Languages:** C, Python, Go, SQL, Java, HCL

**Technologies:** NumPy, PyTorch, Pandas, OpenCV, Linux, Sockets, Git, AWS, GCP, Azure, K8s, Docker

**Topics:** Data Structures and Algorithms, Object Oriented Programing, Functional Programming, Systems, Consensus Algorithms, Actor Model, Network Protocols, TCP, Cryptographic Algorithms, Machine Learning

## EXPERIENCE

**Center for Atmospheric Particle Studies**

**Pittsburgh, PA**

**Research Assistant**

**May 2022 – Aug 2022**

- Developed a low cost device for measuring PM<sub>2.5</sub> air pollutants collected on a foam tape over several months
- Implemented an image processing pipeline as a cheaper alternative to traditional particle detection machines
- Integrated a system of Arduino and Python scripts to control stepper motors based on continuous CV input.

## PROJECTS

**Fontify (Solo Python Project)** *PIL, Image Processing, De-Noising, Computer Vision*

**Jan 2022 – July 2022**

- Created image processing software in Python to convert handwritten letters into a personalized bitmap font.
- Utilized image processing, edge-detection, noise reduction algorithms to detect pencil writing on paper.

**Concurrent Proxy Server (C)** *Git, HTTP, Sockets*

**July 2023**

- Developed a proxy server in C using p\_threads and fork to handle requests concurrently, anonymize traffic and cache responses. Utilized Unix sockets and a bounded cache following an LRU eviction policy.

**Distributed Backend (Golang)** *Replication, Actor Model, Mailbox/Message Passing*

**Nov 2023**

- Designed and executed a concurrent server to manage the state for a multiplayer game, accessible via API.
- Handled client requests about and updates to the game state using RPCs and a message-passing model.
- Implemented node launching and server groups, ensuring replication and enforcing consistency within groups.

**Poker-Bots Hackathon Dev Team** *GCP, K8s, GitHub Actions*

**Mar 2024; Mar 2025**

- Helped CMU Data Science Club run their first AI Poker-Bot competition, with \$6,000 in prizes and 63 teams.
- Used GitHub Actions to automatically build Docker images of user-submitted Python bots, allowing competitors to use custom dependencies and machine learning libraries of their choice, running containers on GCP.
- Helped build the second iteration of the competition in 2025 using AWS ECS for bots, Lambda for matches

**x86 IA-32 Kernel from Scratch** *C, ASM, Simics*

**Aug 2024 – Dec 2024**

- Built a complete i386 kernel from scratch, solo, for CMU i5410, implementing preemptive multitasking
- Engineered hardware interfaces, memory management, and I/O system for concurrent ELF binary execution