Alexander Broihier

847-596-0390 | adb12@illinois.edu | https://alex-d-b.github.io | https://www.linkedin.com/in/alex-broihier88

Education

University of Illinois at Urbana-Champaign **Expected May 2026**

Bachelor of Science Mathematics, Statistics and Computer Science; James Scholar GPA: 4.0/4.0

ACM: SIGma Co-Lead (Math and Algorithms Club)

Relevant Coursework: Data Structures, Algorithms, System Programming, Distributed Systems, Compilers

Technical Skills

Programming Languages: C, C++, Java, JavaScript, TypeScript, Python, Rust, Ocaml
Frameworks/Tools: Git, Pandas, React, Next.js, SQL, MongoDB, Redis, Spring Boot, Docker, Linux

Experience

Box – *Software Engineering Intern* May 2024 – August 2024

- Implemented distributed event processing using **Apache Helix** and **Zookeeper** to split work over 10+ processes and delivered 6 weeks ahead of schedule
- Engineered a configurable framework in **Java** and **Redis** to concurrently process 100,000+ events per second
- Provided an internal events service with **OpenAPI** and **Spring Boot** deployed with **Kubernetes** on **GCP**
- Introduced 8 tracked metrics along with unit and integration tests for 9 classes, uncovering 3 bugs in existing code

IBM – *Accelerate Program: Software Development Track* June 2024 – July 2024

- Studied software design topic such as full stack development and application security with IBM leaders
- Collaborated in groups of 10+ participants to progress through weekly coding projects

Compilers Research January 2024 – Present

- Apply compiler methodologies to data science to increase performance of exploratory data analysis workflows
- Construct novel benchmarks for dataframes, uncovering over 600x performance differences between libraries
- Future Work: leverage LLMs to dynamically generate pattern matching rules

Undergraduate Course Assistant January 2023 – Present

- Create and maintain online **C++** problems in **Docker** application to assess course knowledge of 800+ students
- Execute lab sections and office hours to help students learn course concepts and use **GDB** and **Valgrind** to debug

PeopleWeave Research Project April 2023 – January 2024

- Automated collection of authorship data in **Python** with **Parsel** to power models developed by other teams
- Utilized AWS and multithreading to bolster development workflow and data scraper performance (5x speedup)

iD Tech Instructor June 2023 – August 2023

- Taught robotics engineering and **C++** programming through a course partnership with BattleBots
- Managed classes of 10+ students along with weekly logistics

VEX Robotics Competition Lead Designer, Builder, and Programmer August 2018 – May 2022

- Leveraged **CAD** in **Autodesk Inventor** to speed up the design process, getting robots in the field 3+ weeks faster
- Introduced teammates to version control using **Git**, collectively saving 6+ hours

Project Highlights

Large Compiler (*OCaml*) May 2024 – Present

- Created a compiler to machine code in **OCaml** to implement a statically typed Algol like language
- Utilized various graph algorithms to accomplish tasks and optimizations such as efficient register allocation

Docker Clone (*Go*) June 2024

- Constructed a lightweight **Docker** like containerization service in **Go** to isolate processes run on **Linux** machines

Resume Parser + Personal Website Content Generator (*OCaml*) June 2024

- Automated updating my personal website by constructing a parser in **OCaml** to use my resume to generate and update content on my website

Spear Text Editor (*C++*) January 2024 – April 2024

- Created a terminal based text editor leveraging **NCurses** in **C++** for the display
- Efficiently managed file data using a Piece Table data structure for reduced memory usage and faster editing speed

Compiler and Interpreter (*C*) December 2023 – January 2024

- Created a stack-based bytecode compiler and interpreter in **C** to implement an imperative object-oriented language
- Implemented bytecode optimizations to speed up common use cases for method calls (7x speedup)
- Designed around single pass compilation to ensure performance and enable use as a REPL interpreter

Bus Trip Planner (*Python, Django, TypeScript, React, Next.js, MySQL*) November 2023 – December 2023

- Allowed users to create, share, and rate updatable trips through **Next.js** front-end
- Crafted a **Django API** back-end to interface with a **GCP** hosted **MySQL** instance, efficiently retrieved bus route data
- Leveraged **Google Maps API** to render routes on an interactive map, providing an intuitive user interface

Physics Engine + Personal Website (*Rust, TypeScript, Svelte*) July 2023 – August 2023

- Created a physics engine to power a unique background display for a personal website
- Ensured the physics engine remained performant by compiling **Rust** code to **web assembly**
- Implemented a **CI/CD** pipeline using **GitHub Actions** to automatically build and deploy the website

News Aggregator (*Python, TypeScript, React, Next.js, MongoDB*) May 2023 – July 2023

- Implemented a data scraper in **Python** with **Parsel** to automatically gather and store current news information
- Leveraged **OpenAI API** to AI generate a daily welcome message based on gathered data of current events
- Provided a front-end **React** and **Next.js** app allowing users to search through and view 125+ news articles per day
- Utilized **Google Cloud** as an identity provider to implement secure authentication

SPIMbot (*MIPS ASM*) April 2023

- Wrote **assembly** code for SPIM MIPS Simulator to move a virtual bot and complete various collaborative tasks
- Leveraged **memory mapped IO** to move and read sensor data, allowing the bot to respond to environmental changes
- Implemented coroutines in **MIPS assembly** to speed up solve queens algorithm by 15x

Multiplayer Connect Four App (*Rust*) November 2022 – December 2022

- Implemented a front-end web app with **Yew** framework, providing 3 game modes and 2 AI opponents
- Managed multiplayer lobbies over **TCP** by leveraging **Tokio** as an asynchronous multithreaded runtime
- Included foreign feature interface for existing **C++** code to bolster the back-end server with cheat detection
- Used **GitHub Actions** to automatically build and deploy the web app when code is pushed to GitHub

VEX Robotics Competition Robot Control Codebase (*C++*) February 2021 – May 2022

- Designed a **JavaScript** simulation to test autonomous robot motion algorithms, saving 10+ hours
- Developed then iterated upon the structure and API of a real-time, multithreaded, object-oriented **C++** codebase, allowing teammates to quickly specify advanced, accurate autonomous movement
- Documented the codebase in the team's engineering notebook and through numerous comments in the code
- Achieved the highest programming skills score at the state competition for Illinois in 2022 with 36.25% more points than second place

Database Relevant Link Finder (*Python*) October 2021

- Created a script to receive numerous command line arguments including with desired keywords and optional flags
- Parsed HTML to extract links and find relevant keywords on the pages they linked to
- Improved performance and reduced HTML requests by caching data for future use

3D Maze Generation Algorithm (*C#, Unity*) April 2021 – May 2021

- Created a randomly generating maze in 3D space utilizing a custom algorithm
- Debugged maze generation by leveraging preprocessor macros to compile additional components to alter maze generation

Asteroids Game (*JavaScript*) September 2020 – December 2020

- Guided project before adding additional functionality
- Added a menu system that allowed vast customization of gameplay features, appearances, and difficulty
- Implemented an AI to autonomously fly around and accurately shoot asteroids on the menu screen