

# Kai Tamkun

(831) 419-6171 | [kai@tamkun.io](mailto:kai@tamkun.io) | [linkedin.com/in/ktamkun](https://linkedin.com/in/ktamkun) | [github.com/heimskr](https://github.com/heimskr)

## EXPERIENCE

---

### Software Engineer

October 2024–August 2025

*Bun*

*San Francisco, CA*

- Worked on Bun, a JavaScript runtime that embeds JavaScriptCore
- Improved support for native Node.js modules by implementing Node-API support
- Enhanced compatibility with existing JavaScript ecosystem by implementing missing features in Node modules

### Compiler Engineer

April–August 2024

*SAPEON*

*Santa Clara, CA*

- Developed a compiler in C++ to convert Torch MLIR to a proprietary graph representation to enable PyTorch models to be run on AI inference accelerator hardware
- Parsed MLIR, translated instructions into nodes in a DAG
- Ran passes to optimize and split/merge nodes as necessary
- Serialized the DAG to a Protobuf object

### Teaching Assistant

September–December 2022, April–June 2023

*University of California, Santa Cruz*

*Santa Cruz, CA*

- Managed assignments and handled grading for CSE 150, a course about computer networks, and CSE 13S, an introductory computer science course about C programming

### IT Support

Summer 2020

*Georgiana Bruce Kirby*

*Santa Cruz, CA*

- Helped a local prep school with its IT tasks in preparation for the upcoming school year

## EDUCATION

---

### University of California, Santa Cruz

Santa Cruz, CA

*Master of Science, Computer Science and Engineering*

*June 2023*

### University of California, Santa Cruz

Santa Cruz, CA

*Bachelor of Science with Honors, Computer Engineering*

*June 2020*

## PROJECTS

---

### Untitled Game | C++, OpenGL, GLFW

May 2022 – Present

- Custom OpenGL-based engine and user interface
- Procedurally generated worlds
- Asynchronous network I/O
- Embeds V8 for scripting support
- Makes use of complex multithreading

### JavaScript Engine (Master's Project) | C++, Flex, Bison

2023

- Implementation of a JavaScript interpreter with garbage collection

### Why | C++, Flex, Bison

2017–2023

- Specification for an instruction set architecture
- Implementations of an assembler, linker and virtual machine

### LL2W | C++, Flex, Bison

2019–2023

- Compiler that converts LLVM intermediate representation into Why assembly

### C+- | C++, Flex, Bison

2022

- Compiler for a custom C++-like language that targets Why

## TECHNICAL SKILLS

---

**Languages:** C++23, C, JavaScript, Python, Java, Chisel, Verilog, HTML5, CSS, C#, PHP

**Software:** OpenGL, JavaScriptCore, Bun, V8, Node.js

**Developer Tools:** Git, Linux, Meson, Docker, VS Code, IntelliJ

**Spoken Languages:** English, German