

Gary Huang

mail.gary.huang@gmail.com ■ GitHub.com/GaryHuan9
LinkedIn.com/in/mail-gary-huang ■ U.S. Citizen

EDUCATION

University of Michigan, Ann Arbor, MI

Bachelor of Science in Engineering – Computer Science (GPA: 3.995/4.0)

Apr 2025

Relevant Coursework: Computer Architecture, Probabilistic Methods in Engineering, Parallel Computing, Intro to Operating Systems, Computational Linear Algebra, Multivariable & Vector Calculus, Entrepreneurial Creativity.

SKILLS

Programming: C++, C, .NET C#, SystemVerilog, Python, Java, OpenCL, CUDA, Makefile
Software Platforms: Unity, Blender, Git, Linux, ARM, RISC-V, JetBrains Tools, VS Code, Ghidra
Foreign Language: Native in Chinese

WORKING EXPERIENCES

Computer Science & Engineering Division, University of Michigan

Instructional Aide (Teaching Assistant)

Aug 2023 – Present

- Teach *Data Structures & Algorithms* course lab section to review and clarify lecture content for 35 students.
- Guide students through difficult **C++** debugging processes, fostering effective problem-solving strategies.
- Explain complex programming concepts, instrumental in improving student comprehension of course topics.

Caesar Creek Software, Miamisburg, OH

Software & Reverse Engineering Intern

May 2023 – July 2023

- Performed security vulnerability analysis on *Google Nest Protect* smart home device for the potential of gaining remote code execution through its automatic software update process.
- Examined hardware components including an *ARM Cortex M4* processor to obtain control of device firmware.
- Reverse engineered firmware and identified software packages such as *FreeRTOS* and *lwIP* using *Ghidra*.
- Developed custom **C/C++** toolchain for firmware manipulation; decreased testing time from 30 to 2 minutes.

Information and Technology Services, University of Michigan

Information Assurance Intern

May 2022 – Aug 2022

- Reduced peak load of automated searches in *Splunk* from 30,000 to 4,000 actions per minute per year by designing a heuristic-based greedy optimization algorithm to improve temporal distribution of searches.
- Streamlined custom packaging and deployment of security software *CIS-CAT* with automated **Bash** scripts.

ORGANIZATION & MAJOR PROJECTS

Echo, Physically Based 3D Rendering Software

([GitHub Link](#))

Founder & Lead Maintainer

Oct 2020 – Present

- Develop a multithreaded CPU path tracer from scratch using **C#** capable of synthesizing photorealistic images.
- Research and adapt methods including multiple importance sampling as described in 14 academic papers and book *Physically Based Rendering* (Pharr et al.) to minimize visual variance caused by Monte Carlo integration.
- Implement optimization techniques in **C#** such as acceleration structures to improve performance by 20x.
- Recruit, train, and supervise voluntary contributors online to assist with *Echo* open-source development.
- Discovered, investigated, and reported behavioral discrepancy bug in official **.NET** runtime compiler ([Issues Link](#)).

Michigan Computer Graphics, Student Organization

([Website Link](#))

President & Co-founder

Apr 2023 – Present

- Applied and founded organization to rectify absence of a computer graphics (CG) presence on campus.
- Interviewed professors for suggestions on establishing and managing an academic student organization.
- Design informative lectures and hands-on workshops to engage and introduce members to CG concepts.
- Promote organization to student population, recruiting more than 150 interested members in two weeks.

Computer Architecture Capstone, Register-Transfer Level (RTL) Processor

([Report Link](#))

Group Project Leader

Mar 2023 – Apr 2023

- Organized and led team to build, test, and optimize a *MIPS R10K* style out-of-order processor implementing open-source *RISC-V* ISA using **SystemVerilog** and *Synopsys Design Compiler* under a six-week time constraint.
- Designed advanced processor modules including a fully associative non-blocking data cache and a two-way associative instruction cache with prefetcher which lowered average cycles per instruction from 4.8 to 2.1.
- Identified and optimized critical paths reported by synthesis tools, decreasing clock period from 32 to 7.5ns.