Kevin Huai

kevinhuai.com | huaikevin15291@gmail.com | linkedin.com/in/kevin-huai | github.com/KevinH15291

Education

University of Alberta **May 2027 (expected)**

B.Sc. Honors in Computing Science; Dean's Honor Roll; GPA: 3.70/4.00 *Edmonton, AB*

Relevant Coursework: Compiler Design, Computer Architecture, Algorithms, Operating Systems, Linear Algebra, Graph Theory

Experience

Teaching Assistant - CMPUT 229 **Sept 2025 – Present**

University of Alberta *Edmonton, AB*

- Introduced students to RISC-V assembly and computer architecture concepts.

Undergraduate Research Assistant **May 2025 – August 2025**

Compiler Design and Optimization Laboratory (CDOL) *Edmonton, AB*

- Authored systems/architecture assignments using the gem5 simulator and RISC-V
- Reproduced GPU accelerator behavior from AMD HIP binaries, implemented CPU branch predictors, and demonstrated the Spectre V1 vulnerability in the gem5 simulator.
- Prototyped PyTorch operators with a Triton-based DSL; developed kernels and microbenchmarks to provide an outside perspective to the functionality of the language.

Projects

Game Boy Emulator Web App | *C/C++, SDL3, WASM, Emscription, Javascript, HTML*

- Built a fully functional Game Boy emulator and deployed it to the browser as WebAssembly using Emscripten, using SDL3 + JavaScript for graphics, audio, and controls.
- Implemented a cycle-accurate Sharp SM83 CPU capable of executing the SM83's modified Intel 8080/Zilog Z80 instruction set and interrupts to match the behavior of the original Game Boy
- Implemented a MMU (Memory Management Unit) using public Game Boy technical references; this maps different memory address regions for the cartrige ROM, RAM, and memory mapped IO registers
- Reverse-engineered the PPU (Picture Processing Unit) to render to an application using the SDL3 GUI framework; used additional features in SDL3 to process keyboard inputs to joypad MMIO
- Validated behavior against real ROMs including Pokemon Red/Blue and Tetris, playing Pokemon Blue and performing the MissingNo glitch with no issues or crashes

ECS Rendering Engine | *C++, CMake, Vulkan, OpenGL, SDL3*

- Designed a modular engine with separate Vulkan and OpenGL backends behind a single abstract RenderDevice API. Implemented a complete Vulkan rendering path mirrored in OpenGL.
- Built mesh/model and texture loaders, uniform buffer updates, and a consistent resource interface across backends.
- Integrated a practical ECS architecture with a lightweight physics layer.
- Used modern C++ (perfect forwarding, variadic templates, type traits, smart pointers, STL) to ensure safety and performance.

Event Management App | *Java, Android, Firebase*

- Collaborated in a group with 5 other people to implementing UI designs and mockups from Figma into a functioning Android app.
- Built the models and views with Java/XML for the MVC design pattern; persisted event and user data in Firestore (NoSQL).
- Implemented a background service for push notifications and timely updates.
- Created and integrated unit tests using Expresso UI tests.

Technical Skills

Languages: C/C++, Python, Java, CUDA/HIP, HTML, CSS, Typescript/Javascript, Assembly, Bash
Technologies: CMake, Linux, MacOS, AWS, Firebase, Git/GitHub, gem5
Frameworks: LLVM, Pytorch, OpenGL, Vulkan, SDL, Svelte