Grace Jin

San Jose, CA | (408) 750-7200 | gsj33@cornell.edu | gracejin.dev | github.com/gracejinsotrue | linkedin.com/in/grace-jin-cornell

EDUCATION

Cornell University, College of Engineering — Expected May 2027

B.S. Computer Science, Minor in Artificial Intelligence

- **Relevant Coursework:** Systems Programming, Physically Based Rendering*, Compilers in C++*, Computer Graphics, Machine Learning*, Data Structures and Algorithms, Digital Logic and Computer Organization (*Spring 2026)
- **Organizations:** Grace Hopper Celebration 2025, Cornell XR Project Team Co-Founder, Rewriting the Code, WICC

EXPERIENCE

Software Engineering Intern | Cepton Technologies, San Jose, CA — May 2025 – Aug. 2025

- Built a real-time WGPU-accelerated 3D visualization pipeline on Linux systems to simulate LiDAR data rendered over synthetic autonomous vehicle driving geometries at 60+ FPS
- Implemented WGSL compute shaders to perform parallel raycasting for ML model training data collection
- Built a 3D environment reconstruction system in Rust to generate simulator depth maps integrated within GPU pipeline
- Optimized Rust legacy code to use standard transformation techniques such as matrix SVD and affine transforms, reducing computation time by 40% in internal benchmarks
- Authored engineering blog post documenting this graphics pipeline: developer.cepton.com/blog/AR_simulator

Lead Software Developer | Cornell Center for Teaching Innovation, Ithaca, NY — Oct. 2024 – Present

- Develop 3D Unity visualizations of Gauss's Law and EM waves for a 500+ student electromagnetism course with 2000+ playthroughs, funded by a grant for exemplary educational projects
- Support 30+ students prototyping AR/VR apps on Meta Quest, Unity and Snap OS, resulting in 10+ new project launches within 4 months

Software Developer | Cornell People and Robots Teaching and Learning (PoRTaL), Ithaca, NY — Aug. 2024 – May 2025

- Implemented 20+ training tasks with PDDL, improving LLM reasoning with complex and asynchronous tasks
- Engineered a PyGame interface to generate thousand-line JSON specs for an LLM planning benchmark, cutting significant manual programming time for team of 12

PROJECTS

Ray Tracer in Python [GitHub] | Python, NumPy, PIL, Blender — Nov. 2025 – Dec. 2025

- Wrote a CPU-based Whitted-style ray tracer with Möller-Trumbore intersection, Lambertian shading and glass refraction, capable of rendering hundreds of .obj models with texture mapping and bilinear filtering
- Implemented BVH acceleration with SAH partitioning and multi-core tile-based rendering via ProcessPoolExecutor, achieving 3-5x speedup for 100K+ triangle scenes and reducing total render time by 60%

Graphics Rasterizer Engine - Custom 3D Rendering [GitHub] | C++, SDL2, CUDA, Blender — Jul. 2025 – Present

- Build a custom graphics renderer from scratch with a multi-pass rendering pipeline capable of rendering 50k+ vertices at 30+ FPS, with no external graphics Libraries
- Implement a 3D engine with SDL2 to bridge user input with hierarchical object editing and animation
- Migrating to CUDA for GPU parallelization with a current 5x performance improvement over base CPU implementation

NeuroScent - MIT Reality Hack Hardware Track Winner [DevPost] | C#, Unity, OpenBCI, Arduino — Jan. 2025

- Led team of 5 to develop an immersive VR olfactory biofeedback system and Galea EEG data processor for mental well-being enhancement, won out of 400+ competitors
- Integrated Unity to render calming scenes and trigger Arduino-controlled diffusers upon detecting abnormal biofeedback

RISC-V Pipelined Processor | SystemVerilog, RISC-V ISA — Oct. 2024 – Dec. 2024

- Implemented a 5-stage pipelined processor with hazard detection, stall, squash, and bypass logic
- Wrote SystemVerilog test benches to unit test submodules and ensure accurate instruction execution
- Simulated processor with Quartus simulation tool to efficiently debug processor issues

Computer Science Content Creator [Instagram] — Aug. 2019 – Present

- Built an audience of 18K+ followers and 3M+ video views by posting computing topics, personal projects and digital art

SKILLS

Computer Languages: Python, C++, C, Rust, Typescript, Java, HTML, CSS, ARM Assembly

Web Technologies & Frameworks: WGPU/WebGPU, OpenGL/WebGL, Vulkan, WGSL, Three.js, React, Django

Development Tools: Linux, Git, CUDA, GCC, GDB, RTOS, Docker, Gradle, CI/CD, Unity, Figma, Blender