Evan Luo

+1 (646) 523-5194 | Santa Barbara, CA | evanluo@cs.ucsb.edu | github.com/EvanLuo42 | linkedin.com/in/ziyunluo

Education

University of California, Santa Barbara Sep 2025 — May 2029
Bachelor's of Science, Computer Science *Santa Barbara, CA*

- Cumulative GPA: 4.0/4.0
- Relevant Coursework: Linear Algebra, Single variable & Multivariable Calculus, Discrete Mathematics, Computer Organization and Logic Design, Data Structure and Algorithms

Skills

- **Programming Languages**: **CUDA, C/C++, CUDA, Rust, C#, Python, Java, Typescript, Lean**
- **Graphics Framework**: **Vulkan, OpenGL** and **Metal**; Real-time rendering frameworks including **Falcor, Donut and Mitsuba 3**
- **Graphics Areas**: **Geometry Processing**, **Physically-based Rendering**
- **Game Engine**: **Unity**, **Unreal Engine 5**

Work Experience

Internship Jan 2026 — Apr 2026
Beijing Freedo Technology *Remote*

- Develop Differentiable Physics Simulation & Differentiable Rendering pipeline for Video Generation of Fluid Dynamics with Taichi (e.g. FluidNexus-like pipeline).

Undergraduate Research Assistant Oct 2025 — Present
Four Eyes Lab, UC Santa Barbara *Santa Barbara, CA*

- Investigate algorithms for real-time 3D scene reconstruction and semantic segmentation in XR environments, focusing on multi-view feature aggregation, open-vocabulary scene queries, and efficient spatial representations for interactive reasoning.
- Reproduced works including Open-Vocabulary Online Semantic Mapping

Undergraduate Research Assistant Sep 2025 — Dec 2025
ArchLab, UC Santa Barbara *Santa Barbara, CA*

- Studying **GPU architecture** and **CUDA C++ programming**, with focus on performance modeling and warp-level scheduling.
- Led a collaboration with Bionic Vision Lab to simulate and verify GPU-based vision processing architectures by GPGPU-Sim/Accel-Sim. Designed automated profiling scripts and validated GPU/CPU co-simulation against experimental data.
- Reproduced ISCA 2025 **Forest: Access-Aware GPU UVM Management**, modeling forest-based dependency graphs and analyzing page migration latency.
- Reproduced the performance bottlenecks reported in ISCA 2025 paper **Avant-Garde: Empowering GPUs with Scaled Numeric Formats**, showing how software-managed multi-level scaling in FP8/MX/HBFP formats causes significant instruction and register overhead on current GPUs.
- Implemented RV32I in PyRTL.

Projects

Contributor, Blender Jan 2026 — Present

- Contributed to the Grease Pencil module, implementing and debugging layer/group traversal and UI integration across C++ core and Python RNA APIs. (Pull Request 1, Pull Request 2)
- Investigated and fixed issues related to Grease Pencil tree structures, including node ordering, parent–child relationships, and iteration correctness.
- Worked with Blender's RNA system and UI layout code to expose Grease Pencil data structures to Python and editor panels.

Graphics Programmer, N3R J-RPG Game Oct 2025 — Present

- Designed and implemented a custom Toon render pipeline using Unity Scriptable Render Pipeline (SRP) with RenderGraph-based frame architecture.
- Built a hybrid lighting system with forward rendering for main directional light (shadow ramp + lightmap) and deferred lighting for additional point lights.
- Implemented stylized NPR shading including ramp-based toon lighting, screen-space outlines, and stylized fog.
- Developed modular post-processing passes (bloom, tone mapping, outline) as RenderGraph passes.
- Implemented GBuffer layout and managed render pass dependencies and transient GPU resources.
- Profiled and optimized GPU performance using RenderDoc, NVIDIA Nsight Graphics, and Unity Frame Debugger.
- Trained a hand-written rune recognizer for gameplay using PyTorch + ResNet with dataset generator (automatically generate variants from handwritten rune obtained from a PyGame canvas window).

Maintainer, Marching Cube Terrain (github.com/EvanLuo42/marching-cube/) Jul 2025

- Implemented a real-time voxel terrain sculpting tool based on Marching Cubes with density falloff brush.
- Designed screen-space ray picking with inverse projection and view matrix reconstruction.
- Integrated dynamic mesh rebuilding, brush visualization, and deltaTime-scaled continuous editing.
- Built with C++, GLFW, ImGui, and GLM; supports FPS-style camera with UI debug overlay.

Maintainer, PyRTL RV23I (github.com/EvanLuo42/rv32i/) Oct 2025 — Present

- Implemented a RISC-V RV32I CPU using PyRTL, describing the datapath and control logic entirely in Python.
- Designed modular components including ALU, register file, decoder, sign-extension unit, branch unit, and memory interfaces.
- Verified instruction correctness through unit tests and simulation, covering arithmetic, logic, load/store, and branch instructions.