

# Albert An

510-894-5186 | [betoan808@gmail.com](mailto:betoan808@gmail.com) | [linkedin.com/in/beto-an](https://www.linkedin.com/in/beto-an) | [betoan.com](https://betoan.com) | U.S. Citizen

## EDUCATION

### The University of Texas at Austin

GPA: 3.90

*M.S. Computer Science, Integrated B.S./M.S. Program*

*Aug. 2025 – May 2027*

*B.S. Computer Science, **Turing Scholars Honors Program***

*Aug. 2022 – May 2026*

*B.S. Mathematics*

*Aug. 2022 – May 2025*

## COMPETITIONS & SKILLS

**Awards & Qualifications:** USAMO (2021), USAJMO (2020), USAPhO Honorable Mention (2021), USAPhO (2020)

**Languages:** C, C++, Python, Java, C#, Lua, TypeScript, JavaScript, SQL, GLSL, Bash, Lean 4

**Technologies:** PyTorch, Hugging Face, QLoRA, GRPO, pandas, Polars, scikit-learn, .NET, React, FastAPI, WebGL2, OpenGL

**Tools:** Git, Docker, CMake, Kubernetes, Jenkins, GDB, Valgrind, NVIDIA Nsight, RenderDoc, LeanDojo, Claude, MCP

**Selected Coursework:** Machine Learning, AI, OS, RL, NLP, Computer Vision, Graphics, Physical Simulation, ML Systems

## EXPERIENCE

### Roblox

May 2026 – Aug. 2026

*Software Engineer Intern | Game Engine, Audio Team*

*San Mateo, CA*

- Developed Roblox’s unified wind-audio system and integrated it into the Audio API graph, converting object-relative wind and physics-driven turbulence into real-time spatial audio without prerecorded samples
- Designed turbulence, foliage, and whistling synthesis layers with smooth parameter modulation driven by surrounding physics and geometry, producing natural, stochastic, non-repetitive ambient sound
- Built geometry-aware occlusion with adaptive raycasting, adjusting ray counts based on compute budgets and estimated motion of occluding geometry relative to the sound target
- Achieved per-instance costs of **5  $\mu$ s for mixer-thread DSP**, **10  $\mu$ s for worker-thread logic** (physics, preprocessing, and raycasting), and **under 1 KB of memory**
- Owned zero-to-one development with Claude and MCP tools in Roblox Studio; shipped after performance and stress testing

### The University of Texas at Austin

Aug. 2025 – Present

*Graduate Researcher | Computer Graphics Laboratory*

*Austin, TX*

- Developed a real-time frame-interpolation system that reprojects motion vectors and pixel history from multiple prior frames to exploit long-horizon temporal coherence
- Trained a lightweight confidence-based blending network to learn per-pixel visibility and reliability weights, resolving occlusions and motion disagreement under strict latency constraints

### IMC Trading

Jun. 2025 – Aug. 2025

*Quantitative Trader Intern | D1 NOII*

*Chicago, IL*

- Built an automated Python trading system for a simulated options and futures exchange, implementing quoting and taking strategies, delta hedging, position management, and counterparty-aware execution
- Trained a scikit-learn ridge-regression strategy based on NOII signal arrival order; built an event-driven execution backtester to model fills and slippage, then analyzed multi-exchange results with pandas and Polars

### Wells Fargo

Jun. 2023 – Aug. 2023 | Jun. 2024 – Aug. 2024

*Software Engineer Intern | Enterprise Technology, Risk Reporting*

*Dallas, TX | San Francisco, CA*

- Built a .NET and C# productivity dashboard with recommendations for frequently used applications; deployed the suite across company laptops and Azure and WAVE virtual desktops
- Created a synthetic UI testing pipeline using Login Enterprise, Microsoft Intune, and Ansible Tower, enabling application teams to execute and report automated compatibility tests

## PROJECTS

### Networked Car-Soccer Simulation [\[link\]](#) | C++, raylib, ENet, UDP, Premake

- Developed a playable 3D Rocket League-style game and custom rigid-body engine with 120 Hz fixed-step simulation, quaternion dynamics, broad-phase and narrow-phase collision detection, and an iterative sequential-impulse contact solver
- Unified local, server, and client execution around a shared simulation core; built authoritative multiplayer with tick-stamped redundant inputs, 30 Hz snapshots, local prediction, and reconciliation-based correction

### Lean 4 Theorem Prover | Python, PyTorch, Qwen 8B, LeanDojo, GRPO, QLoRA

- Fine-tuned an 8B-parameter Qwen model for Lean 4 theorem proving using a 4-bit base model and 16-bit LoRA adapters, training only 43M parameters (0.5%) with supervised fine-tuning followed by Group Relative Policy Optimization (GRPO)
- Built a LeanDojo verified-reward pipeline; on 250 held-out problems, baseline/SFT/SFT+GRPO achieved **0%/13.2%/17.6%** proof success and **0/1.5/1.8** average valid proof steps

### Portal Rendering Engine [\[link\]](#) | TypeScript, WebGL2, GLSL

- Implemented recursive multi-pass portal rendering with stencil masking and depth management for nested portal views; integrated instanced drawing, first-person physics, and teleportation that preserves velocity across portals

### Minecraft Voxel Engine [\[link\]](#) | TypeScript, WebGL2, GLSL

- Built infinite chunked voxel terrain with lazy loading, hysteresis-based retention, Perlin-noise generation, procedural shaders, a day-night cycle, and collision physics

### Model Rig [\[link\]](#) | TypeScript, WebGL2, GLSL

- Created an interactive skeletal-animation editor with quaternion-based hierarchical manipulation, four-bone linear-blend skinning, keyframe interpolation, GPU thumbnails, and JSON serialization