

Albert An

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

USAMO (2021)

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

USAJMO (2020)

Technologies: PyTorch, pandas, Polars, scikit-learn, Hugging Face, Lean 4, .NET, React, FastAPI, WebGL, OpenGL

USAPhO Honorable

Tools: Git, Docker, CMake, Kubernetes, Jenkins, Ansible, GDB, Valgrind, NVIDIA Nsight, RenderDoc

Mention (2021)

Selected Coursework: ML, AI, OS, RL, NLP, CV, Graphics, Physical Simulation, ML Systems

USAPhO (2020)

EXPERIENCE

Roblox

May 2026 – Aug. 2026

Software Engineer Intern | Game Engine, Audio Team

San Mateo, CA

- Developed Roblox’s unified wind-audio system, a real-time procedural-synthesis framework that converts object-relative wind and physics-driven turbulence into responsive spatial audio without prerecorded samples
- Designed three synthesis layers for turbulence, foliage, and whistling, with **all audio procedurally synthesized** to support smooth parameter modulation and geometry-derived foliage and whistle profiles
- Integrated into Roblox’s Audio API graph and added geometry-aware occlusion to duck wind audio behind obstacles
- 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 [\[link\]](#) | 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; GRPO raised proof success 4.0 percentage points to 17.6% (44/250), 61% of GPT-4’s 29.0%, using an 8B model with 0.45% as many total parameters as GPT-4’s estimated 1.76T

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

- Implemented recursive multi-pass portal rendering with stencil and depth management, instanced drawing, first-person physics, and velocity-preserving teleportation

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