

Mason Liu

(469) 732-7075 | liumasn@gmail.com | [linkedin.com/in/masonliu](https://www.linkedin.com/in/masonliu) | github.com/masonliu | masonliu.dev

EDUCATION

The University of Texas at Dallas
Bachelor of Science in Computer Science

Aug. 2025 – May 2029
Dallas, TX

EXPERIENCE

Research Software Engineer | Python, C#, C/C++, .NET, Unity
Multimodal Interaction Lab, University of Texas at Dallas

Jan. 2026 – Present
Richardson, TX

- Engineered vibrotactile calibration pipelines, C#/Unity environments, and real-time wearable interfaces for Google-funded YouTube haptic research and Samsung C&T warehouse training.
- Led Python-based processing and analysis for a CHI 2027 paper, handling hundreds of millions of hand-joint coordinates from 120 Hz motion capture and synchronized RGB footage while correcting and filtering 50,000+ malformed capture frames.
- Built the Unity/Meta Quest multiplayer runtime for UIST and ISMAR 2026, integrating BLE event transport with GPIO/PWM and H-bridge actuation; achieved 16 ms median event-to-actuation latency with <0.05% dropped events across 2,000+ hardware-in-the-loop cycles.

Full-Stack Engineer | JavaScript, React, Node.js, Express, PostgreSQL
Mase Labs (Freelance)

Oct. 2025 – Jan. 2026
Dallas, TX

- Built backend order workflows for a commerce and fulfillment platform, integrating checkout verification, queue-backed processing, retry handling, and status tracking.
- Supported 5,000+ fulfilled orders per month with 99.97% fulfillment success and sub-10s median order-creation-to-queue latency.
- Reduced release overhead by cutting manual deployment steps from 6 to 2 and average deploy time from 25 minutes to 8 minutes.

PROJECTS & OPEN SOURCE CONTRIBUTIONS

Lutris | Python, GTK, PyGObject

2026 | 10.2k GitHub stars

- Spearheaded a 3-path installation methodology with sub-10ms local handoff in Python/GTK, shipping to Lutris' 835,500+ active users.
- Hardened installer API handling for the third-party ModDB Python package source, preserving a seamless user experience with <0.035% request-failure and configuration-error paths.

Hindset | C#, .NET, Avalonia

2026

- Developed Hindset, a C#/.NET performance-analysis tool that turns 75+ users' screen, input, and timing patterns into concrete, context-aware workflow feedback.
- Built persistent per-user model memory with cross-modal embeddings and neural sequence modeling, tailoring profiles and analytics across 10,000+ analyses while reducing redundant findings by 85%.
- Hardened the review state machine with atomic replacement, exclusive locking, idempotent revisions, and deletion fencing; passed 5,000+ fault-injection scenarios with zero stale or duplicated results.

Bottles | Python, GTK, PyGObject

2026 | 8.8k GitHub stars

- Reworked Bottles' native lifecycle by centralizing stable launch-ID generation and cleanup across client-side state entries, shipping to Bottles' 3,400,000+ monthly active users.
- Prevented deprecated launcher state from leaking across clients by wiring teardown hooks into program and bottle deletion flows, with regression tests spanning 4 deletion paths and reducing stale-entry failures by 99.93%.

TECHNICAL SKILLS

Languages: Python, C#, JavaScript, TypeScript, Java, C/C++, HTML/CSS

Frameworks & Libraries: .NET, Unity, React, Node.js, Express, Avalonia, GTK, PyGObject, Meta Quest SDK, Octokit, NumPy, pandas, SciPy, Zod

Tools & AI Platforms: Git, Linux, OptiTrack, GitHub Actions, Docker, SQLite, PostgreSQL, Redis, Stripe, Electron, OpenTelemetry, OpenAI API, Anthropic API, Gemini API, Ollama, llama.cpp, ComfyUI, MCP