

Matthew Hamilton

matthewhamilton3141@gmail.com | [linkedin.com/in/matthewhamilton3141](https://www.linkedin.com/in/matthewhamilton3141) | github.com/matthewhamilton3141 | matthewhamilton.dev

EDUCATION

University of Waterloo

Waterloo, ON

Bachelor of Applied Science in Systems Design Engineering

Sep. 2026 – May 2031 (Expected)

- **Relevant Coursework:** Digital Computation (SYDE 121, C++), Linear Algebra (MATH 115)

PROJECTS

Retermina | *Tauri v2, Rust, React, TypeScript*

2026 – Present

- Built a local-first desktop terminal workspace on Tauri v2; a Rust backend spawns and multiplexes native PTY sessions per pane over Tauri IPC, with no cloud backend
- Engineered a draggable 12-column multi-panel canvas (react-grid-layout) with collision-aware layout — split terminals, a syntax-highlighted editor, and a project-wide live git diff — that persists per folder across the last 30 projects
- Turned it into a supervision layer for coding agents: embedded Claude Code with a live context-window gauge (parses session JSONL to chart context fill against the model window) and per-project token and cost accounting
- Shipped a GitHub Actions release pipeline building universal (Apple Silicon + Intel) macOS binaries and a minisign-signed in-app auto-updater that verifies each download against a `latest.json` manifest before applying

gsplat-rt | *Python, TensorRT, CUDA, PyTorch, OpenUSD, Isaac Sim*

2026

- Built a real-time monocular reconstruction pipeline converting live RGB video into a 3D Gaussian Splat scene and physics-ready collision mesh (OpenUSD export for NVIDIA Isaac Sim / Omniverse); a multithreaded lock-free architecture sustains 82.7 FPS end-to-end (12.1 ms/frame) on an NVIDIA A10G, run in NVIDIA NGC containers (`--gpus all` via the NVIDIA Container Toolkit, volume-mounted model caches) with host-driver/container-runtime debugging
- Compiled a strongly typed FP16 TensorRT engine for Depth Anything V2 with zero per-frame GPU allocations, cutting depth latency to 6.3 ms/frame (2.24× faster than TF32 at >0.9999 output correlation)
- Deployed a SuperPoint + LightGlue learned SLAM front-end as an FP16 TensorRT engine at 7.4 ms/frame (18× over ONNX Runtime's CUDA EP), cutting trajectory error to 3.5 cm ATE on TUM RGB-D (vs. 5.7 cm for classical ORB)
- Wrote a custom CUDA kernel for TSDF fusion at 0.06 ms/frame — under 0.5% of the pipeline's 12.1 ms frame budget — feeding marching-cubes mesh extraction and PhysX collision export
- Trained a PPO navigation policy (Gymnasium / stable-baselines3) on the reconstructed scenes; a one-step-lookahead safety shield drove it to 0 collisions in evaluation while beating a hand-tuned baseline on path efficiency, verified by a physics drop-test PASS in Isaac Sim 5.1

Sketchstack | *Next.js, React, TypeScript, React Flow, Supabase, Tailwind CSS*

2026

- Built a visual systems-design canvas that compiles a hand-drawn architecture diagram into a structured, mode-aware Markdown prompt for AI coding agents; guest-first, with everything but cloud save working without an account
- Engineered an interactive graph editor on React Flow (@xyflow/react) — 50+ typed components with per-type icons, connect/reconnect edges, multi-select align/distribute with snapping guides, undo/redo, and a keyboard-driven command palette
- Wrote a prompt generator that traverses the graph into a grouped spec (merging bidirectional connections), with swappable node packs and templates across four diagram modes: System Design, App/UI Flow, Database Schema, and Task Planning
- Built a Supabase backend — GitHub OAuth auth and row-level-secured Postgres for cloud-saved diagrams plus one-click read-only share links with rich previews; deployed on Vercel

TECHNICAL SKILLS

Languages: TypeScript, Python, Rust, C++, JavaScript, HTML/CSS

Frameworks & Libraries: React, Next.js, Node.js, Tauri, Tailwind CSS, PyTorch, OpenCV, stable-baselines3, Gymnasium, PyBullet

Tools & Platforms: Git, Linux, CUDA, TensorRT, ONNX, Docker, NVIDIA Container Toolkit, NGC, OpenUSD, NVIDIA Isaac Sim, Vercel

AI / ML: 3D Gaussian Splatting, Depth Anything V2, reinforcement learning (PPO), visual SLAM, TensorRT engine optimization