# Tyler Flar

San Diego, CA | +1-619-569-3700 | tylerflar@gmail.com | linkedin.com/in/tylerflar | tylerflar.com

## Education

- **University of California – San Diego** La Jolla, CA
  *Master's, Computer Science* *Apr 2025 – Dec 2026*
- **University of California – San Diego** La Jolla, CA
  *Bachelor's, Computer Science; GPA: 3.8* *Sep 2021 – Mar 2025*

## Professional Experience

- **Lycanworks** Remote
  *Freelance Artist* *Jan 2026 – Present*
  - Run a commission-based 2D art service (icons, portraits, full-body, ref sheets) with clearly defined pricing, scope, and revision policies; manage client intake via Discord/Instagram and deliver web-ready + full-res assets.
  - Authored Terms of Service covering payment milestones, delivery guarantees/refunds, and copyright/usage to protect both client IP and artist rights.
  - Built and maintained the Lycanworks portfolio/commission website and social presence to market work, showcase examples, and streamline requests.
- **UC San Diego Advanced Robotics and Controls Lab** La Jolla, CA
  *Graduate Student Researcher* *Sep 2025 – Present*
  - Developing a real-time monocular SLAM pipeline for in-body 3D reconstruction (kidney/bladder) to enable autonomous steering of a flexible surgical robot.
  - Built a real-time robotic simulation pipeline in NVIDIA Isaac Sim to evaluate monocular SLAM and navigation algorithms in anatomically constrained environments, with physics-based collision modeling and camera pose supervision.
- **Engineers for Exploration** La Jolla, CA
  *Project Lead* *Jan 2023 – Oct 2025*
  - Led two wildlife-tracking projects and mentored rotating student teams (5–15 active at a time; 20+ cumulatively) building low-cost animal-monitoring systems.
  - Drove end-to-end delivery across sensing hardware, onboard compute, DSP/computer-vision pipelines, and field-ready ground tooling.

  ***Radio Telemetry Tracker*** *Dec 2024 – Oct 2025*
  - Modernized a drone-based wildlife radio-telemetry platform that detects OOK VHF collar transmitters (∼138–235 MHz) to replace manual on-foot triangulation with automated aerial surveys.
  - Implemented a unified SDR + DSP stack as a pip-installable Python package backed by C++ (FFTW/Boost/UHD via pybind11) to stream IQ, detect pings, and emit real-time callbacks.
  - Added multi-SDR hardware support (USRP B200mini, AirSpy, HackRF) behind a shared interface with consistent buffering, timestamping, and lifecycle controls.
  - Built an OOK ping detector using FFT-based channelization, dynamic thresholding, and width/SNR gating; produced raw IQ captures and CSV ping logs for reproducible offline analysis.
  - Implemented RSSI-based transmitter localization with a path-loss model and SciPy least-squares optimization, including bounds and frequency-specific ping aggregation.
  - Built a cross-platform Ground Control Station (PyQt6/QWebEngine + React/TypeScript + Leaflet) with offline map tile caching, scan controls, and amplitude-encoded ping overlays.
  - Implemented a reliability-focused control link using framed protobuf packets with CRC, acknowledgments/retries, timeouts, and connection-quality metrics; added hardware-free simulator modes.
  - Delivered tower power-system analysis and documentation, including an energy budget, solar sizing from irradiance data, and LTspice hot-swap safety simulations.

  ***Baboons on the Move*** *Jan 2023 – Dec 2024*
  - Built and benchmarked aerial-video baselines for detecting small wildlife in drone footage, including labeling coordination and quality checks for the Laikipia Baboon Aerial Dataset.

- Implemented an adaptive multi-Gaussian background subtractor in C++/OpenCV with first-N-frame initialization, bounded-variance updates, and shadow suppression; added a CLI for step-through debugging and batch video export.
- Recreated a published MATLAB motion-detection pipeline that combines multi-frame differencing with low-rank background modeling (nuclear-norm optimization); fused motion masks, applied morphology cleanup, and tracked detections across frames with temporal consistency and interpolation.
- Built a reproducible hyperparameter search harness with checkpointing and optional parallel execution, saving intermediate masks and overlays for fair comparisons and ablation studies.

- **Leidos** Remote
  *Chief Engineer Intern* *May 2025 – Sep 2025*
  - Led a DOJ monitoring revamp: evaluated monitoring tools, authored a finance-backed white paper, and rolled out probes + ServiceNow alerts—saving $120K/year and raising critical portal uptime to 99.9%.
  - Automated ATF-mandated RCA follow-ups with a C# parser that extracts action items from outage docs, builds a live Excel status tracker, and pushes tasks via REST to ServiceNow.
- **UC San Diego, CSE Department** La Jolla, CA
  *Teaching Assistant* *Mar 2024 – Sep 2025*
  - Co-taught UCSD's core Algorithms sequence for 100+ students; designed walkthroughs on graphs, greedy/DP, and divide-and-conquer.
  - Delivered 1:1 and group tutoring; earned consistently Strongly Positive TA evaluations.
- **Cubic Transportation Systems** San Diego, CA
  *Systems Engineering Intern* *Jun 2023 – Sep 2023*
  - Automated requirements traceability with a Python + DXL tool that pulls DOORS requirements and TestRail results into a live matrix, cutting ∼1 month of manual cross-checking to ∼30 minutes.
  - Streamlined analytics for the San Francisco Clipper Card back office via parameterized SQL scripts, enabling clients to resolve 80%+ of recurring data requests instantly.

## Projects

- **Jackdaw**
  *Balatro simulator for reinforcement learning research* *Mar 2026 – Present*
  - Built a complete Python reimplementation of the Balatro game engine (∼30 modules, all 150 joker effects, bit-exact 3-layer PRNG matching LuaJIT 2.1) exposed as a Gymnasium-compatible RL environment.
  - Designed entity-based observations (235-dimensional) with a factored action space (21 action types) and trained agents using Stable-Baselines3 MaskablePPO.
  - Implemented a validation bridge against live Balatro gameplay and ∼250 CLI scenarios to verify deterministic engine correctness.
- **Yeen GPT**
  *Small GPT language model from scratch* *Feb 2026 – Feb 2026*
  - Built and trained a small GPT language model end-to-end in PyTorch, including a custom Byte-Pair Encoding (BPE) tokenizer and dataset batching pipeline.
  - Implemented a Transformer decoder (multi-head self-attention, positional embeddings, residual blocks) plus checkpointed training and a CLI for training and text generation (CPU/CUDA).
- **Stoatix**
  *Benchmarking and diagnostics CLI* *Dec 2025 – Dec 2025*
  - Built a lightweight benchmarking/diagnostics CLI that runs YAML-defined suites with matrix expansion, warmups/runs/retries, timeouts, env/cwd overrides, and optional CPU pinning.
  - Implemented an audit-friendly results pipeline (session metadata, resolved config, cases manifest, JSONL per-iteration records) and per-case summary stats (median/mean/p95/stdev) with configurable IQR outlier filtering.
  - Added run-to-run comparison to classify regressions/improvements/added/removed cases with noise heuristics (CV, p95/median, minimum OK iterations) for CI regression detection.
  - Integrated Linux perf tooling (perf stat counters; optional perf record + FlameGraph SVG generation) with graceful fallback; automated CI baselines via GitHub Actions artifacts and gating workflows.

- **Personal Website**
  *Content-driven portfolio site with Eleventy* *Dec 2025 – Dec 2025*
  - Built a fast, content-driven personal site with reusable Nunjucks components (card grids, timeline), JSON-backed data, and Markdown posts/pages.
  - Implemented enhanced Markdown rendering with MathJax v3 (inline + block LaTeX) and Prism.js syntax highlighting, including custom language aliases (asm, SystemVerilog).
  - Added custom Eleventy filters (readable dates, collection filtering) and computed "back link" navigation based on page tags for consistent UX across blog/classes/projects.
  - Set up GitHub Pages deployment with an Eleventy build pipeline (outputs to docs/), static asset passthrough, and metadata for SEO/social previews (Open Graph/Twitter cards).
- **Ibexicon**
  *Wordle-style game solver* *Aug 2025 – Aug 2025*
  - Built a Wordle-style assistant that recommends next guesses using a composite score: expected information gain + solve probability, with dynamic exploration/exploitation weighting.
  - Optimized solver throughput via Web Worker scoring, precomputed binary pattern tables, and optional Rust/WASM acceleration for fast feedback pattern computation.
  - Added shareable session URLs that encode the current board state (no secrets stored), plus privacy-first optional telemetry and CI (Vitest + Playwright) with automated GitHub Pages deploy.
- **FollowMe Robot**
  *Bluetooth direction-finding for Boston Dynamics Spot* *Apr 2025 – Jun 2025*
  - Built a Bluetooth direction-finding "follow-me" stack for Boston Dynamics Spot using BLE angle-of-arrival plus RSSI-based ranging to estimate 3D tag position in real time.
  - Implemented the host pipeline in C (NCP/BGAPI ingest to AoA solve to spherical-to-Cartesian) and published pose + angle telemetry over MQTT.
  - Integrated the telemetry stream into ROS and added filtering (averaging, outlier rejection, smoothing) to stabilize tracking under multipath and RF noise.
  - Submitted manuscript to IEEE/ASME International Conference on Advanced Intelligent Mechatronics.
- **Music Genre Predictor**
  *Multi-class genre classifier* *Sep 2024 – Dec 2024*
  - Built an end-to-end multi-class genre classifier on a Spotify audio-feature dataset (15,150 tracks, 19 genres), including EDA, feature engineering, model selection, and evaluation.
  - Addressed class imbalance with SMOTETomek and improved feature behavior using log transforms on highly skewed features plus standardized scaling.
  - Tuned models with GridSearchCV and selected a soft-voting ensemble reaching ∼0.376 test accuracy; analyzed per-genre precision/recall via confusion matrices.
- **MineCord Link**
  *Minecraft–Discord bridge plugin* *Jul 2024 – Sep 2024*
  - Built a Spigot/Paper plugin that bridges server chat and gameplay events to Discord via a bot + webhooks (join/leave, deaths, advancements, start/stop notifications).
  - Implemented Discord slash commands to run server commands, list online players, share in-game coordinates, and toggle cross-chat; enforced administrator + allowlist checks.
  - Added a short-lived linking-code flow to connect Discord accounts to Minecraft UUIDs, enabling enriched usernames/mentions when relaying messages.
  - Shipped operational features including YAML config migration safety, per-listener enable/disable controls, and an auto-update checker that pulls the latest GitHub release.
- **Basic Needs Center Study**
  *Food insecurity research at UCSD* *Jan 2024 – Mar 2024*
  - Analyzed UCOP Basic Needs Dashboard trends alongside a custom UCSD survey (n=116, 22 variables) to evaluate how service awareness vs utilization relates to reported food insecurity.
  - Cleaned and standardized inconsistent survey schemas across years, reconciled identity labels, and merged datasets for subgroup comparisons.

  - Built EDA dashboards and quantified relationships with Pearson correlations, finding weak associations between service awareness and food insecurity.

- **Cache Simulator**
  *N-way set-associative CPU cache simulator* *Aug 2023 – Aug 2023*
  - Simulated an N-way set-associative CPU cache with configurable address width, cache size, line size, and associativity to study locality and cache behavior.
  - Implemented LRU-style eviction using per-line timestamps and tracked total accesses, hit rate, miss rate, and per-category miss counts.
  - Classified misses as compulsory, conflict, or capacity by tracking first-touch behavior and archiving evicted lines across sets.
  - Generated a per-access CSV report (address, tag/index, hit/miss, miss type) and printed a summary breakdown for quick analysis.
- **Fortune Telling with Panda Express**
  *Interactive browser game* *Apr 2023 – Jun 2023*
  - Built an interactive browser game in HTML/CSS/JavaScript that guides users through menu selections and reveals a fortune with sound effects and a typewriter-style animation.
  - Implemented a data-driven “fortune engine” and nutrition calculator that aggregates dish macros and displays Nutrition Facts with percent daily values.
  - Set up continuous integration with GitHub Actions to auto-run Prettier/ESLint formatting and Jest unit tests on pushes and pull requests; published JSDoc developer documentation.
- **RoboCup SSL Robot Runtime**
  *Raspberry Pi runtime for omni-drive robot control* *Sep 2022 – Jun 2023*
  - Built a Raspberry Pi runtime that subscribes to AI UDP multicast commands, decodes Protobuf messages, and converts velocity targets into 4-wheel omni-drive RPM setpoints.
  - Streamed motor commands to an STM32 controller over UART and added telemetry receive/parsing with header filtering and buffer resets.
  - Implemented optional analytics tooling to plot expected vs actual wheel velocities in real time and export graphs for tuning/debug.

## Leadership & Volunteering

- **Triton Tails (Furries at UC San Diego)**
  *Founding President* *May 2023 – Jun 2025*
  - Founded and led a university-recognized student org; authored the org handbook (governance, officer/principal-member processes, event planning, funding/TAP workflows, and moderation policies).
  - Designed and programmed the public-facing website end-to-end using 11ty + Vite + Bootstrap/Sass, adding interactive pages (events calendar, team, store, galleries) and maintainable content/data templates.
  - Set up GitHub Actions CI for lint/format/tests/HTML validation and automated deploys to GitHub Pages.

## Coursework

- **Statistical Natural Language Processing** *Jan 2026 – Mar 2026*
  - Built end-to-end NLP models in PyTorch, from bag-of-words and Deep Averaging Networks to custom transformer encoder/decoder architectures for classification and language modeling.
  - Implemented Byte Pair Encoding (BPE) from scratch and evaluated vocabulary/embedding initialization tradeoffs on sentiment analysis.
  - Developed generation + RAG components: beam search decoding for GPT-2 and a FAISS/LangChain retrieval-augmented QA pipeline over meeting transcripts (QMSum).
- **Algorithm Design and Analysis** *Sep 2025 – Dec 2025*
  - Built a graph isomorphism alignment heuristic: Weisfeiler–Lehman color refinement to unique-color seed matches to seed-and-extend greedy mapping to local swap improvements within color classes, with Hungarian-based optimal reassignment for small classes.
  - Achieved 3.95 score (∼96% correct alignment) on the course benchmark; completed 20 advanced LeetCode problems (graphs, DP, flows, randomized, NP-completeness).

- **Parallel Computation** *Sep 2025 – Dec 2025*
  - Hand-tuned DGEMM on AWS Graviton3: cache-blocking + packed panels with an ARM SVE microkernel; matched/slightly beat OpenBLAS on the target instance.
  - High-performance GPU GEMM with CUDA on NVIDIA T4: shared-memory tiling + register micro-tiles, warp-friendly layouts, light autotuning; achieved cuBLAS-class throughput (∼4.8 TFLOP/s on large matrices).
  - Distributed 2D stencil/wave simulator on SDSC Expanse using MPI: 2D domain decomposition, nonblocking halo exchange, global reductions; strong scaling to hundreds of cores.
- **Introduction to Robotics** *Sep 2025 – Dec 2025*
  - Programmed a skid-steer rover in ROS 2 on Ubuntu, using camera + IMU for autonomous navigation.
  - Built a waypoint-following controller, upgraded it from open-loop to vision-based closed-loop landmark localization, and implemented EKF-SLAM to jointly estimate robot trajectory + landmark map in a 10×10 ft arena.
  - Designed and justified two motion planners (safety-first vs time/distance-optimal) and performed SLAM-based mapping in an obstacle field.
- **Principles of AI: Probabilistic Reasoning and Learning** *Apr 2025 – Jun 2025*
  - Built small probabilistic systems including a Bayesian Hangman solver (letter prediction via word-frequency priors), an HMM + Viterbi decoder for hidden-state sequence inference, and a movie recommender using a latent user-type model trained with EM.
  - Applied EM to learn a noisy-OR medical diagnosis model (SPECT heart dataset), tracking log-likelihood and classification error, and completed proofs/derivations in information theory (entropy, KL-divergence, mutual information).
- **Programming Languages** *Jan 2025 – Mar 2025*
  - Implemented a small Haskell-like language ("Nano") with an interpreter + REPL, including parsing/lexing, closures, recursion, lists, and built-ins; later extended evaluation with exception semantics (throw/try-catch) using Either.
  - Built supporting functional-programming projects: $\lambda$-calculus reductions and encodings, tail-recursive/fixed-point utilities, random-art expression trees (AST + evaluator), and arbitrary-precision BigInt arithmetic using folds and explicit carry logic.
- **Introduction to Computer Security** *Sep 2024 – Dec 2024*
  - Performed 32-bit x86 binary analysis with GDB on Linux, tracing stack frames, registers, and control flow.
  - In controlled lab targets, demonstrated common vulnerabilities—buffer overflows, timing/memory side channels, and web attacks (SQLi/XSS/CSRF)—and connected each to standard mitigations (parameterized queries, output encoding, CSRF tokens).
- **Web Mining and Recommender Systems** *Sep 2024 – Dec 2024*
  - Built Python recommenders on real user–book interaction data: (1) rating predictor using global mean + user/item bias terms with L2 regularization and convergence checks, and (2) read/not-read predictor using interpretable signals (popularity bands, user activity, prior reads).
  - Exported predictions to CSV for hidden-set evaluation; achieved top 3% RMSE (∼1.5) on ratings and ∼76% accuracy (top 24%) on read prediction.
- **Artificial Intelligence: Searching and Reasoning** *Apr 2024 – Jun 2024*
  - Studied core AI methods including classical search, adversarial search, reinforcement learning, bandits & Monte Carlo Tree Search, constraint solving, and propositional/first-order reasoning.
- **Principles of Computer Operating Systems** *Jan 2024 – Mar 2024*
  - Built foundational OS concepts through C labs in UMIX (user-space teaching OS): processes/threads, context switching, CPU scheduling, and synchronization primitives.
  - Reinforced core systems topics: virtual memory & allocation, filesystems, and I/O behavior/tradeoffs.
- **Introduction to Computer Architecture: A Software Perspective** *Aug 2023 – Sep 2023*
  - Analyzed compiled x86 kernels to model performance: counted dynamic instructions, categorized memory/ALU/branch ops, and computed cycles/runtime under different CPI/clock/ISA assumptions; applied Amdahl's Law to parallel speedup with overhead.
  - Simulated cache behavior (tag/index/offset derivation, LRU hit/miss tracing, miss-rate breakdown) and virtual memory/TLB scenarios (address translation, TLB misses vs. page faults, LRU updates).

  - Estimated CPI/ILP across in-order vs. OoO pipelines (dependencies, resource limits) and compared multicore vs. SMT scheduling plus memory-consistency outcomes using fences/barriers.

- **Software Project for Computer Architecture** *Aug 2023 – Sep 2023*
  - Built and analyzed microbenchmarks using hardware performance counters (perfstats) to measure IC, cycles, CPI, and ET; validated predictions with the performance equation and Amdahl's Law.
  - Optimized real kernels by improving branch predictability, cache/TLB behavior (pointer-chasing "miss machines" for MLP), applying loop tiling/unrolling for convolution, and implementing parallel histograms with locks vs. per-thread reduction using OpenMP (including hyperthreading/ILP/MLP experiments).
- **Digital Systems Laboratory** *Jul 2023 – Aug 2023*
  - Built and simulated digital hardware in SystemVerilog, progressing from combinational/sequential modules to full systems (microcoded multiplier, alarm clock/calendar, traffic-light FSMs, LFSR stream cipher).
  - Designed datapaths + control units, timed FSMs, and modular counters/register files; validated with self-checking testbenches and edge-case timing/state verification.
- **Components and Design Techniques for Computer Systems** *Jul 2023 – Aug 2023*
  - Developed fundamentals in Boolean algebra, logic minimization (K-maps, SOP/POS), and translating between truth tables, expressions, and logic diagrams.
  - Designed and analyzed sequential circuits (latches, D flip-flops, timing diagrams) and implemented finite state machines from state diagrams/transition tables using flip-flops + combinational logic, with exposure to basic processor/control/memory design concepts.
- **Software Engineering** *Apr 2023 – Jun 2023*
  - Practiced team-based dev workflow with Git/GitHub (branches, PRs, issues/labels, standups) and GitHub Pages deployment.
  - Built responsive sites with semantic HTML + CSS (flexbox/grid, media queries) and debugged using Chrome DevTools.
  - Implemented JavaScript features including web components (Shadow DOM), localStorage, timers, and robust error handling/monitoring.

## Skills

- **Languages**: Python, C/C++, C#, JavaScript/TypeScript, SQL, Haskell, SystemVerilog, MATLAB, Java
- **AI/ML**: PyTorch, scikit-learn, NumPy, SciPy, OpenCV, LLMs (training, fine-tuning), RAG (FAISS/LangChain), CUDA
- **Web & App**: React, Node.js, HTML/CSS, Eleventy, Bootstrap/Sass, PyQt6, Web Components
- **Systems & Hardware**: ROS 2, MPI, OpenMP, ARM SVE, Protobuf, UART, BLE/MQTT, NVIDIA Isaac Sim
- **Tools & Platforms**: Git, GitHub Actions CI/CD, Docker, AWS (EC2), Linux, ServiceNow, Microsoft 365