

Alexander Rowe

🐙 github.com/aprowe | 📍 San Diego, CA

SUMMARY

Software engineer with 8+ years building simulation systems, real-time graphics, and developer tools in C++, Rust, Python, and TypeScript. Experience spans large-scale orbital simulation on HPC clusters, GPU rendering pipelines in OpenGL and WebGL, distributed data processing, and interactive visualization tools used daily by engineers and analysts.

EXPERIENCE

SpaceX

May 2021 – December 2023

Software Engineer II

- Developed **C++ simulation software** (Bazel, Slurm/HPC) running large-scale orbital planning cycles that used evolutionary algorithms to search for optimal mission trajectories.
- Abstracted the **minimizer interface** and optimized the underlying search algorithms, working with gravitational models, orbital mechanics, propellant budgets, and quaternion math.
- Introduced a **modular component system** — drawing on functional programming and ECS design — letting features and tests be selectively enabled or disabled, and modernized a monolithic configuration format to JSON.
- Sole author and maintainer of a **browser-based 3D visualization tool** (TypeScript, WebGL, Lit, Redux, Webpack) rendering the Earth, trajectories, cones, regions, and points of interest with custom shaders, a scrubbable timeline, and a collapsible entity tree.
- Kept visualization state consistent using one-way data flow, Redux selectors, and diffed updates so stateful WebGL side effects never drifted out of sync; added screenshot capture, single-file HTML export with embedded data, and a Flask service that rendered screenshot sequences into video.
- Built a documentation site generated by converting **TypeScript types into JSON Schema** through a custom build tool, hosting released versions, examples, and API docs internally.
- Built a **real-time distributed simulation dashboard** using Kafka, Protobuf/gRPC, PostgreSQL, and Python/C++ services for configuring runs and streaming results as they arrived.
- Worked on a team of 5–8 with high individual ownership, operating like a startup within the company and presenting work directly to stakeholder groups.

Senseye

November 2017 – October 2020

Lead Engineer (first engineer on team)

- Built eye-tracking analysis software for detecting neurological and cognitive signals, starting with **C++/Qt** prototypes on top of Pupil Labs hardware and moving to production Python systems.
- Designed a **Python data processing pipeline** for high-frequency physiological sensor data using **Dask**, NumPy, SciPy, and TensorFlow, usable both for offline model training and as a real-time online algorithm.
- Delivered an Air Force research project correlating pupil data with flight telemetry captured through the **P3D simulator API** as pilots flew simulated missions.
- Designed and deployed a **distributed real-time streaming API** using Kong as an API gateway, Docker Swarm for orchestration, and Protobuf for inter-service communication.
- Wrote experiment-running and data-collection tooling that correlated eye data with eye-camera video, including compression and batch processing of very large video datasets.
- Set up **CI/CD** and nightly data runs with Jenkins and Buddy; hosted internal libraries on a private PyPI server.
- Hired and managed engineers and established the code review practices and engineering standards as the company grew from 4 to ~30 people.

Synesthesia

2016; August 2025 – Present

Software Engineer

- Cross-platform (Windows/macOS) live visual performance application for artists and VJs, rendering real-time interactive shaders to an **OpenGL** canvas.

- Led a major refactor of the **C++ graphics engine** to a data-driven **command pattern** with immutable models and diff-based change tracking, separating static structure from per-frame runtime state.
- Implemented multi-pass rendering, real-time compositing and blend modes, cubemaps, and GPU resource management (texture pooling, dynamic resizing without full pipeline rebuilds) across AMD and Nvidia hardware.
- Built the audio pipeline mapping frequency data to GPU textures for audio reactivity, and integrated real-time video sharing via **Spout, NDI, and Syphon**.
- Built the **React/Redux** front end (React Router, Webpack, Ant Design, SCSS grid layouts) for controlling generative visuals.
- Embedded the **Duktape** JavaScript engine and exposed C++ APIs to a scripting layer; wrote JSON and custom shader-format (SSF) serialization with file migration and versioning, covered by **Catch2** unit tests.

PROJECTS

Evolution Simulation

- Evolutionary algorithm engines implemented in **Rust**, JavaScript, and Python, with interactive visualizations of population dynamics over time.

Rust Graphics and Tooling

- Real-time rendering and developer tool projects in **Rust** using **Vulkan** and **egui**, focused on performance-sensitive interactive applications.

EDUCATION

University of California, Santa Cruz

2010 – 2014

B.S. Physics

Santa Cruz, CA

- 2nd Place, UCSC Hackathon, 2014 & 2015
- Computational physics thesis using a gravitational model written in Python.

SKILLS

Languages : C++, Rust, Python, TypeScript/JavaScript, GLSL, Objective-C, SQL, HTML/SCSS

Graphics : OpenGL, WebGL, Vulkan, GLSL shaders, multi-pass rendering, GPU resource management, Mapbox

Simulation & Scientific Computing : Orbital mechanics, evolutionary algorithms, HPC/Slurm, NumPy, SciPy, Dask, TensorFlow

Frontend : React, Redux, Lit, Webpack, Ant Design, interactive data visualization

Systems & Infrastructure : Docker and Docker Swarm, Kafka, Protobuf/gRPC, PostgreSQL, Bazel, Jenkins, GitHub Actions, Kong

Practices : Modular and functional architecture, test-driven development, profiling and optimization, technical mentorship