Anthony Green

Greater Seattle Area, WA

anthony.j.green@outlook.com | (253) 495-2988 | linkedin.com/in/anthonygreen03 | github.com/gusjengis

Systems engineer specializing in Rust, WebGPU, GPU compute, and WASM for portability and performance.

EXPERIENCE

Backend Engineer | C#, Python, JS — **05/2025 - Present**

Rainspire Studios — Seattle, WA

- Led the integration of mobile titles with cloud services.
- Handled user authentication, cloud storage, serverless code, and ad integration.
- Built gameplay features and tooling in Unity (C#).
- Managed iOS builds in Xcode and managed Apple Developer account.
- Prototyped adaptive bots using Unity ML-Agents to automate balance testing as mechanics evolved.
- Implemented in-engine data visualization to accelerate debugging and performance analysis.

Pharmacy Technician — **08/2024 - 04/2025**

Walgreens — Puyallup, WA

Undergraduate Researcher | Rust, WebGPU, Python — **09/2023 - 12/2024**

University of Washington — **github.com/gusjengis/Physics-Sim**

- Lead developer of a physics simulator for earthquake simulation.
- Built engine from scratch using Rust and WebGPU; simulation and rendering all done on the GPU.
- Built extensive tooling for experiment setup, runtime control, measurement, visualization, and automated analysis.
- Collaborated with faculty to align technical design with research and performance needs.

PROJECTS

hyprlog | Rust — **github.com/gusjengis/hyprlog**

- Used Rust to create a Linux service for hyprland that logs window focus events. First activity tracker for the platform.
- Created a termianl UI that uses these logs to render an interactive activity/screen time report.
- Setup a CD/CI pipeline using Github Actions to automatically publish releases to several package managers.

scutl | Rust — **github.com/gusjengis/scutl**

- CLI tool that generates mermaid diagrams of projects
- Uses the Language Server Protocol to build accurate diagrams for any language.

Timeline Prototype | Rust, WebGL, WASM, JS

- Created a prototype website using Rust/WASM for the business logic.
- Used WebGL from Rust for high performance rendering of huge quantities of data.
- Used a traditional JS canvas layer for well-styled UI.

Terrain Generator | Rust, WebGPU, WASM, JS — **portfolio.agreenweb.com/perlin**

- Implemented Perlin noise from scratch using WebGPU compute shaders.
- Stacked layers of this noise in rendering shaders to generate 3D terrain with lighting and camera controls.
- Used WASM and JS to imbed this Rust program in my portfolio site for easy sharing.

Plinth | Rust, WebGPU, WASM, TS, SolidJS — **portfolio.agreenweb.com/perlin**

- Setup a template repo that allows me to quickly get started with an advanced web stack.
- The stack consists of a Rust core via WASM, uses WebGPU for rendering and a Typescript + SolidJS layer for advanced UI.
- This is all set up with a custom dev server that enables hot reloading for rapid iteration.

TECHNICAL SKILLS

Languages: Rust, Python, JS, C64 Basic, C (K&R & ANSI), C++, Arduino, HTML, CSS, WASM, WGSL/WebGPU, GLSL/WebGL, C#, HLSL, Nix, Markdown, Typst

Tools: Neovim, OpenCode(AI Agent), Linux, Git, Unity, Docker, AWS, Windows, MacOS, Arduino

EDUCATION

B.S. in Computer Science - University of Washington (2021 - 2023)
Associates of Science - Pierce College (Running Start) (2019 - 2021)