Sanjay Sankaran

773-669-5131 | sanjayjs@uw.edu | linkedin.com/in/sanjay-sankaran | github.com/zyugyzarc

Education

University of Washington Seattle, WA
Bachelor of Computer Science - Data Science *March 2024 – July 2026*

Bellevue College Bellevue, WA
Associate's in Arts and Sciences *July 2023 – March 2024*

Projects

TraiNNer - Neural Network Designer | *Pytorch, QT, Dataflow*

- Designed and developed an interactive Python application to design neural networks.
- Implemented the runtimes various neural network components and layers.
- Developed visualization for network weights and code-flow for better model and program intuition. beans

Eddie - Reinforcement Learning Robot | *Reinforcement Learning, Pytorch, Godot, Raspberri Pi*

- Built a 2-wheeled robot, and then used reinforcement learning and SLAM to pathfind around an environment.
- Trained the RL-Agent in a simulation built with Godot and Blender, and then fine-tuned it in the real word.
- Implemented and tested various RL methods, including PPO, SAC, and I2A.

Hot Potato - Multiplayer Game | *WebRTC, Decentralized Networks, Redis, React*

- Developed a Decentralized P2P game using a custom protocol built on top of WebRTC.
- Created a custom React-like UI framework to support stateful interactions with zero dependencies.
- Deployed and Used redis for client-to-client handshakes, following a pub-sub architecture.

MiniGPT - Implementation of an SLM | *Multihead Attention, Transformer Language Models, Pytorch*

- Implemented a Small language model with Multihead Attention, and trained it on the Shakespeare text corpus.
- Optimized the model for the best set of hyperparameters.

Blocksim - Hardware Accelerated Physics Engine | *Vulkan, GPGPU, GLSL, GLFW, C++*

- Designed and Built a 2D Hardware Accelerated Physics Engine with the Vulkan C API.
- Parallelized the base physics engine with Compute shaders for framerates upto $\sim$ 5000 FPS.
- Implemented a mix of KD-trees and Spacial subdivision for high volume (100,000-s of objects) collision resolution.

Ascii Render - 3D Graphics Engine | *Python, C++, Curses, Linear Algebra*

- Designed and Developed a 3D raster graphics engine that runs on a terminal console.
- Used tools like curses and ansi codes to interface with the terminal to support graphics.
- Implemented the core features of a graphics pipeline, such as Vertex projection, Z-Buffering, Texture interpolation, Transform Object Models, Material Systems, Per-vertex attributes, Instancing and more.
- Built a high-speed and light core (in C++) while still being easy-to-use through a Python interface.

Relevant Coursework

SWE: CSE 331 - Software Design and Implementation, CSE 332 - Datastructures and Parallelism
Systems: CSE 451 - Operating systems, CSE 333 - Systems programming, CSE 351 - Hardware/Software interface
ML/AI: CSE 446 - Machine learning, CSE 447 - Natural Language Processing
Data Science: CSE 344 - Database Management, CSE 444 - Database Internals, CSE 442 - Data Visualization
Software Design: CSE 331 - Software Design, CSE 401 - Compilers

Technical Skills

Languages: C, C++, Python, JavaScript/Typescript, Java, SQL, Assembly, GLSL (glslc/spv)
Web Development: Fullstack Development, Flask/Django, REST APIs, Websockets, React
Machine Learning: Pytorch, LLMs, Reinforcement Learning, Transformer Networks, Autoencoder Models
Systems: Linux, Docker, Bash, Low-level networking
Embedded Systems / Robotics: FreeRTOS for Arduino, ATTiny, ESP boards, Python for Raspberry Pi
Computer Graphics: Vulkan, GLSL programming - Graphics and Compute, GPGPU
Application Development: GTK and QT with C++ and Python