

# Josh Friedson, Software Engineer

U.S. Citizen based in San Diego, CA

GitHub.com/jfriedson | LinkedIn.com/in/JoshFriedson | jfriedson.github.io | Business@JoshFriedson.com

## PROFESSIONAL EXPERIENCE

---

### **Software Engineer**

Nov 2022 - May 2024

*JPMorgan Chase & Co.*

*Columbus, OH*

- Authored and maintained Spring Boot web apps' functionalities utilizing OpenAPI, Oracle DB, IBM MQ, and Apache Kafka
- Containerized microservices with Docker, managed scaling with Kubernetes, and migrated from a private cloud to AWS
- Contrived JMeter scripts for performance and stress testing of web apps, conducted testing on BlazeMeter, and utilized Dynatrace, Grafana, and Splunk to collect and analyze metrics for performance and cost optimization
- Evaluated app resiliency using Gremlin, BlazeMeter, and ChaosMonkey
- Upgraded JDK and dependencies for Spring Boot apps while retaining functionality and increasing unit test coverage

### **Intern Engineer**

June 2016 - Aug 2016

*Kenautics*

*San Diego, CA*

- Collaborated with San Diego Harbor Police to discover ways of improving handheld diving equipment with the prospect of upgrading the underwater experience for professional divers
- Designed and began development of a prototype Java Android app with navigation capability for an underwater handheld navigation device

## EDUCATION

---

### **BASc, Computer Science**

Aug 2017 - May 2021

*San Diego State University*

*San Diego, CA*

- Successfully completed upper division electives in Artificial Intelligence, Wireless Networks, Computer Security, and Database Theory and Implementation

## OPEN-SOURCE CONTRIBUTIONS

---

### **MNIST Digit Recognition App in Electron + React + Go**

- Electron desktop application uses typesafe IPC to send system info data between the main and render processes
- React provides UI displaying system resource usage and sends an HTTP request containing the MNIST input in the HTML Canvas element provided by the user, displaying the result on response
- Go back end listens for the HTTP request, processes the MNIST input using personally implemented neural network layers testing multiple acceleration strategies utilizing goroutines that have been benchmarked on the github project page, and responds to the request with the inference result

### **Event Ticketing App with Contactless Check-In**

- Event hosts authenticate guests' tickets using Near Field Communication (NFC) on Android devices
- Web server is written in Go (Golang) with the Fiber framework
- PostgreSQL database contains users, businesses, events, and tickets
- Android app developed in Kotlin with React Native

CONTINUED BELOW

## Voxel Grid and Octree Ray Tracer, Parallelized Voxelizer, and Voxel Particle Simulator

- Implemented ray tracing algorithms to quickly and efficiently render voxel (uniformly aligned 3D blocks) grids and octrees consisting of diffuse color, transparency, and normal data efficiently and in real-time using modern C++ and an OpenGL compute shader
- Implemented a voxelization algorithm to convert traditional 3D models made up of triangles into a sparse octree directly on the GPU in a geometry shader
- Designed and implemented a 3D particle simulation where parallelized physics calculations take place in real-time using an OpenGL compute shader, eliminating the need for CPU and GPU memory swap

## Neural Network-Driven Cars Evolved by Genetic Algorithms

- An ensemble of genetic algorithms crossbreed and mutate the best performing neural networks to navigate a car through a race car track in as little time as possible
- The neural network takes 13 inputs: 7 forward-facing distance sensors, and 6 about the car's physical state, such as the velocity and steering angle
- The neural network's output controls the car's accelerator, steering, and standard and emergency brakes
- The project comprises Python for track creation and Javascript for inference, and runs natively in modern web browsers

## UNIVERSITY PROJECTS

---

### RISC Assembly Interpreter in C++

- Interpret SIC/XE assembly source code from a plain text file
- General purpose, program flow, and floating point registers
- All register, bitwise, integer, and floating point manipulation instructions
- Comparison and conditional jump instructions
- Interrupt handling and device interfacing for keyboard input

## CERTIFICATIONS

---

### AWS Certified Developer - Associate

*Amazon Web Services*

Feb 1, 2024

### Certified Kubernetes Application Developer (CKAD)

*The Linux Foundation*

Jan 7, 2023

### Machine Learning Specialization

*DeepLearning.AI, Stanford University*

Sep 11, 2022

### Deep Learning Specialization

*DeepLearning.AI*

Aug 24, 2022

## TECHNICAL SKILLS

---

**Programming Languages:** C, Modern C++, Java, Go, Python, SQL, GLSL

**Frameworks:** Spring Boot, Fiber, Express, React

**Database Systems:** MongoDB, MySQL, Oracle Database, PostgreSQL

**Testing Frameworks and Tools:** JUnit, Mockito, JMeter, ChaosMonkey, Gremlin

**Machine Learning Instruments:** PyTorch, Tensorflow, Scikit-Learn, Stable-Baselines, OpenAI Gym

**Graphics APIs:** OpenGL

**Computer Vision:** OpenCV

**Embedded Systems:** ARM Cortex-M7

**Misc.:** Git, Jenkins, IBM MQ, Apache Kafka