Malakai Spann

MalakaiSpann@gmail.com • (718) 593-1969 • MalakaiSpann.com

A full-performance, devoted software engineer focused on data processing, developing scalable, cross-platform software, and enhancing development processes through formal methods, principles, and modern technologies.

QUALIFICATIONS

Florida Institute of Technology

> Bachelor of Science, Computer Science

Top Secret/Secret Compartmented Information Clearance

> Polygraph & Full-Scope Background Investigation
> January 2024

TECHNICAL EXPERIENCE

Programming

> Languages: Java, Python, C/C++, Zig, HTML/CSS, Java/TypeScript, Go, Rust, Bash

> Libraries/Frameworks: Spring, NumPy, React, Tailwind, PyTest, JUnit

> Tools: Github/Gitlab, Docker, JIRA, AWS, Jenkins

Fundamentals

Strong background in the formal SDLC and Object-Oriented Programming. Also experienced with data processing, embedded systems, software security, developer operations (DevOps), and web-based technologies.

PROFESSIONAL EXPERIENCE

Red Alpha, Columbia, MD
Software Engineer, Data Tradecraft & Artificial Intelligence (AI), Nov 2024 – Current
- Collaborate with leaders and SME across the organization to develop the first Intellectual Property (IP) product.
- Co-lead design, development, and testing of a platform to enable AI-powered workflows.
- Provide support to subcontract missions.

National Security Agency, Fort Meade, MD[1]
Software Engineer, Computer Network Operations, March 2024 – November 2024

- Lead effort to improve development process using DevOps technologies and formal software methods.
- Lead effort to integrate modern build environments for software products.
- Manage technical communications with external product owners to develop enterprise-level solutions.
- Collaborate with a small team to ensure customer satisfaction and product quality.
- Design, develop, and test high-performance data processing, storage, and transmission features.
- Utilize industry standard tools to create/track, prioritize, and resolve customer issues, feature requests, and improvement work.

Lockheed Martin, Denver, CO
Software Engineer Intern, Enterprise Flight Software, May 2022 – Dec 2023

- Collaborate with a small team to develop safety-critical, configurable software for data processing.
- Utilize data transportation protocols (I2C, SRIO, MS1553-B, SpaceWire) to implement a data network across embedded spacecraft devices.
- Participant in customer and business area planning meetings to ensure product goals are achieved within schedule.
- Coordinate with internal and external resources to design, develop, test, and review code.
- Utilize DevOps technologies such as Docker, Gitlab, and various other CI/CD tools to ensure consistent delivery of working code and quality standards.

---

[1] Case number: RES-2024-08400

PROJECTS

- [Hephaestus](): An ever evolving, general-utility Python library. Published to PyPi.

- [HTML Parser](): A purpose-built HTML parser written in Zig – accurate, performant, robust.

- [Adjustable Image Recognition Keyboard System (AirKeys)](): A proof-of-concept, projection-based virtual keyboard utilizing Computer Vision to detect and transmit user input to multiple devices. The project heavily focuses on image recognition, hard/software integration, embedded development, and the development process. Serves as my undergraduate capstone project. Written in Python.
    - o [Project Pitch (Slideshow)]()
    - o [Project Website (In Progress)]()
    - o Development process includes custom docker image/development environment using Visual Studio Dev Containers, custom scripts, and industry standard tools such as Sphinx and PyTest for documentation & quality assurance.

- [SemMed – Neo4j](): A project using the National Library of Medicine's Semantic Medline Database to create a graphical-relational database (Neo4j) using Python.

- [Tic-Tac-Toe w/ AI](): Uses a combination of the minmax (including an alpha-beta pruning version) algorithm and OOP concepts to create a Tic-Tac-Toe game in Python.

- Traveling Salesman Problem (TSP) w/ AI: A set of solutions and test runs for the classic Traveling Salesman Computing Problem using Python and the Genetic & Ant Swarm Optimization Algorithms.
    - ● [Genetic Algorithm Version]() : An implementation of a genetic algorithm that explores the various decision-making methods applicable to the TSP while providing in-depth explanations of the design choices and their effects on the implementation.
    - ● [Ant Swarm Optimization Version]() : An implementation similar to the genetic algorithm version that uses the ant swarm optimization algorithm instead.