JOHN ROSE

johnrose@umich.edu ● Ann Arbor, MI ● johnr282.github.io

EDUCATION

University of Michigan **Ann Arbor, MI**

B.S. in Engineering in Computer Science, B.M. in Cello Performance, Minor in Mathematics *May 2025*

- GPA: 3.91 / 4.00
- Relevant coursework: Distributed Systems, Operating Systems, Machine Learning, Video Game Development, Foundations of Computer Science, Computer Organization, Linear Algebra, Real Analysis

RESEARCH EXPERIENCE

Ordered Systems Lab **Ann Arbor, MI**

LLM-Powered Bug Diagnosis for Distributed Systems *September 2024 - Present*

- Assisted in the development of the Tree of Thoughts prompting strategy used by an LLM agent in order to generate comprehensive diagnoses for given bug reports; written in Python using LangChain
- Modified agent to produce well-structured output that can be compared with ground truth bug diagnoses, using multiple bug location metrics such as file, function, and line to determine diagnosis accuracy

INDUSTRY EXPERIENCE

Dassault Systèmes **Royal Oak, MI**

DELMIA Simulation and Equipment Software Engineering Intern *May 2024 - August 2024*

- Added import and export support to DELMIA for the Unified Robot Description Format (URDF), an XML file format commonly used by ROS applications to describe the kinematics and geometry of robots
- Allowed physical products such as robots and other mechanical equipment to be converted between DELMIA and URDF while maintaining all kinematic (joint locations, limits, types, axes of rotation, etc.) and geometric features
- Gained experience with creating and using complex C++ APIs for manipulating physical products and their associated kinematics, working with XML data, and utilizing robust test frameworks

DELMIA Simulation and Equipment Software Engineering Intern *May 2023 - August 2023*

- Enabled tasks created in DELMIA to be executed on a Doosan-M1013 collaborative robot
- Developed a ROS 2 package in C++ allowing a client to remotely control the physical robot
- Implemented an external motion planner in DELMIA using C++ and created custom communication protocols to send motion commands to the ROS 2 package described previously
- Transmitted the robot motion data back to DELMIA, synchronizing the simulation and physical robot in real time

PROJECT TEAMS

University of Michigan Autonomous Robotic Vehicle Project Team **Ann Arbor, MI**

Navigation Assistant Lead *June 2024 - Present*

- Utilized VMware Hypervisor products to create pre-configured ROS development environments
- Designed and administered an onboarding project to teach the basics of autonomous navigation and ROS application development to over 30 members

Navigation Member *September 2022 - May 2024*

- Developed a resilient, efficient path-planning solution for navigating an autonomous vehicle through an obstacle course using C++, ROS 2, and Nav2
- Built the waypoint navigation system allowing the vehicle to travel to a given series of GPS coordinates
- Utilized ROS 2 to integrate the navigation stack with sensors, computer vision, and embedded systems stacks

PROJECTS

Distributed Systems Projects

Distributed Sharded Key-Value Database with Paxos *November 2024 - December 2024*

- Built a linearizable distributed database that partitions the key-value space into shards; written in Go
- Constructed the system out of many replica groups, each of which is responsible for a subset of shards
- Implemented a configuration service that manages the current set of replica groups and assignment of shards to each replica group, as well as the transitions between assignments
- Configuration service and all replica groups are each fault-tolerant clusters using the Paxos protocol for consensus, provided by a custom-built Paxos library

Online Multiplayer Strategy Game (Work in Progress) *January 2024 - Present*

- Built a state management system on top of the networking library Photon Fusion to ensure game state is properly replicated across all clients and the server and every client request is properly validated; built in Unity with C#
- Created an elegant interface for the state management system that abstracts the complexities of online multiplayer away from the core game logic
- Developed an algorithm for procedurally generating a hex map consisting of equally distributed continents

Operating Systems Projects

Multithreaded Networked File Server *April 2024*

- Developed a file server in C++ allowing connected clients to concurrently create, read, write, and delete files
- Gained experience with file system structure, socket programming, validating potentially malicious client requests, and leveraging reader/writer locks to maximize concurrency while guaranteeing safety and liveness

Thread Library *February 2024*

- Wrote a thread library supporting thread creation, thread joins, synchronization mechanisms such as mutexes and condition variables, and utilization of multiple CPUs; written in C++

Memory Pager *March 2024*

- Developed a system to handle the allocation, storage, retrieval, reading, and writing of memory pages, as well as sharing memory between processes and cleanup of a process's memory when it exits; written in C++

Machine Learning Projects

Dog Breed Classification with Deep Learning *October 2024*

- Utilized several deep learning techniques to classify dog images by breed; written in Python using PyTorch
- Gained experience with convolutional neural networks, transfer learning, and vision transformers

Predicting Survival in the ICU *September 2024*

- Trained a classifier to identify ICU patients at risk of in-hospital mortality using Python and scikit-learn
- Gained experience with data pre-processing, feature engineering, hyperparameter selection for regularized logistic regression models using cross-validation, and kernelized ridge regression models

Miscellaneous Projects

Linear Algebra Library *May 2023 - December 2023*

- Built a fully templated linear algebra library in C++ supporting most standard vector and matrix operations
- Gained experience with numerical algorithms, optimization and data storage decisions required to efficiently work with matrices, writing comprehensive unit tests, creating tools for benchmarking, and writing documentation

Rent-a-Bot *October 2023 - December 2023*

- Developed a cooperative factory simulation game with a team of 3 other students in which players process resources, fabricate and deliver products, and fix various disasters as they occur
- Worked primarily as a gameplay programmer and level designer; built several core gameplay systems such as processors, resources, conveyor belts, item spawners, levers, and timers and designed all but one of the levels
- Won 2nd place out of over 15 games at the end-of-semester showcase; built in Unity with C#

SKILLS

- Languages: C/C++, C#, Go, Python, Java
- Tools: ROS, Unity, Git, PyTorch, scikit-learn, LangChain, VMware Hypervisors, Flask, MongoDB