Dylan Nichols

(845) 707 - 2820 • don5082@rit.edu • linkedin.com/in/dylan--nichols/ • github.com/don5082 • dylannichols.me

EDUCATION

Bachelor of Science in Computer Science, Minor in Math May '28

Rochester Institute of Technology

- GPA: 3.35
- **Relevant Courses:** Machine *Learning, Advanced Linear Algebra, Concepts of Computer Systems, Principles of Data Management, Software Engineering, Mechanics of Programming, University Physics I*

SKILLS

- **Languages:** Python, C/C++, Java, MIPS, HTML, CSS
- **Frameworks/Libraries:** PyTorch, NumPy, Scikit-learn, Angular
- **Tools:** Git, Linux, CI/CD, Vim
- **Concepts:** LLM fine-tuning & training, Neural Networks, HPC

WORK EXPERIENCE

AI Undergraduate Researcher (Large Language Models) Jan '26 - Present

RIT Computer Science Department

- Evaluating the efficacy of SafeDelta in **mitigating safety degradation** by fine-tuning **Llama2-7b** on the **PureBad** dataset using a **High Performance Computing (HPC) Cluster**
- Analyzing safety-utility trade-offs using Attack Success Rate (ASR) and Harmfulness Score (HS) metrics, mapping LLM safety degradation and recovery via SafeDelta.

Lifeguard Apr '23 - Present

Liberty Parks and Rec.

- Managed personnel and resolved escalated stakeholder issues as Acting Pool Director, coordinating facility safety and team operations.
- Executed emergency response protocols in high-risk environments, remaining calm and leading under extreme pressure

PROJECTS

Software Engineering Project Team Leader Jan '25 - May '25

RIT SWEN 261

- Led a 5-person team using **Agile Methodologies** to design and build U-fund, a donation platform- **coordinating deadlines, teamwork,** and **meetings**
- Developed Angular front-end (Landing Page, coherent style) including 5 pages, integrated with Java backend
- Facilitated 4 Agile sprints, each time improving delivery time, coordination, and quality of product, ensuring **zero major bugs** in the final product release

Custom 2D Physics Engine Dec '25 - Jan '26

Python

- Developed a real-time rigid-body engine modeling **gravity** and **environmental boundary collisions** using **vector math and time-step integration** without relying on existing physics libraries
- Engineered deterministic time-step integration logic, allowing the engine to stably compute and render over **10,000 independent entities** without significant performance degradation

Recursive Merge Sort in MIPS Assembly Oct '25 - Oct '25

MIPS Assembly, Vim

- Engineered a high-performance **recursive merge sort** from scratch, demonstrating an advanced understanding of the **MIPS calling convention** and **stack frame management** to handle nested function calls
- Optimized data handling through precise **register allocation** and manual **pointer arithmetic**, ensuring efficient memory access patterns and minimal overhead during the "divide and conquer" execution
- Managed low-level **buffer allocation** and memory-mapped I/O, proving great understanding over the hardware-software interface