Paul Hondola

0745166873 | paulhondola@gmail.com | linkedin.com/in/paulhondola | github.com/paulhondola

Education

Bachelor of Science in Computer Engineering Oct. 2023 – June 2027
Universitatea Politehnica Timisoara

DeepBlue Maker - Underwater Robotics Summer Camp Jul. 2025 – Aug. 2025
Hangzhou Dianzi University, China

Experience

Full Stack Developer Jul. 2025 – current
Hibyte | Typescript, Angular, NestJS, Payload CMS, Supabase, Docker, Github Actions

- GameBox - A full-stack monorepo for managing a game center, featuring a content management system with admin interface and user-facing web application.
- Developed with the help of a modern Tech Stack:
  - Frontend: Angular v20, Typescript, HTML, SCSS
  - Backend & Database: NestJS & Payload CMS & Supabase
  - CI/CD: Docker, Github Actions

Malware Analyst Trainee Apr. 2025 – Jun. 2025
Bitdefender | Java + jadx, C & x86 Assembly + IDA, Python

- Participated in Bitdefender's Academic Labs program, focused on reverse engineering and malware analysis. Working hands-on with Windows and Android environments to analyze vulnerabilities, study malware behavior, and explore exploitation techniques.
  - Studied Android system architecture, its security model and APKs
  - Developed skills in static and dynamic analysis, decompilation and disassembly
  - Reverse engineered simple encryption algorithms inside ransomware

Technical Skills

Programming Languages: C, C++, Python, Java, TypeScript, Bash, SQL
Frontend Development: Angular, HTML5, SCSS
Backend & Databases: NestJS, Supabase, Payload CMS, REST APIs
DevOps & Tooling: Linux, Docker, Git, GitHub Actions, Make, Clang
Languages: English C1 (Cambridge Assessment)

Projects

Benchmark Suite | *Python, psutil, pycpu-info, ML libraries, NumPy, Pandas, Docker* May 2025 - Jun 2025

- Cross-platform system performance benchmarking suite built for in-depth analysis and comparison of CPU, GPU, memory, and cache performance across workloads, architectures, and environments.
- Configurable microbenchmarks: floating point throughput, memory latency/bandwidth, thread scalability
- ML workloads powered by scikit-learn, PyTorch (CPU/GPU/MPS), and TensorFlow
- Compiler benchmarking with gcc / clang via real-world C project compilation
- Detailed hardware info introspection (RAM, CPU cores, frequencies, per-core usage, cache levels)

Treasure Hunt System | *C, POSIX system calls, Clang, Make, Git* Mar 2025 – May 2025

- Introduces an interactive shell-like CLI program to manage hunts and treasures via commands
- Uses logs to track user operations, with symlinked logs for centralized access
- Utilizes multi-process architecture and sigaction-based signal handling for inter-process communication
- Enables runtime features such as live monitoring, hunt and treasure inspection, and controlled shutdown of the monitor process

SafetyMap - Community-driven Safety App | *Java, Android, Google Maps API, Firebase* Nov 2024 – Nov 2024

- Interactive Map: Mark and view safety alerts using Google Maps
- User Alerts: Users can drop pins on the map to report issues such as thefts, road hazards, or other dangers

- Notifications: Real-time notifications for users approaching an area with a safety alert
- Community Trust System: Users can vote on the validity of alerts, contributing to a community trust score

FPGA Video Transmission and Image Processing | *Verilog, VHDL, Xilinx FPGA* Mar 2024 – Jun 2024

- Hardware based video transmission and image processing system, with camera input and display via VGA.
- Supports basic image processing and integrates with OpenCV for face recognition through UART.