Matthias **Bienvenu**

2 Rue Geoffroy Saint-Hilaire, 91000 Évry-Courcouronnes

(+33) 7 83 57 49 98 matthias.bienvenu@telecom-sudparis.eu matthiasbienvenu

Education

Télécom SudParis, Institut Polytechnique de Paris - Engineering degree — Évry, France

2nd year of General Engineering degree in Digital Technologies (Equivalent Master year 1) — 2024 – Present

Lycée Clemenceau — Preparatory Classes (MP2I → MPI*) — Nantes, France

Two-year intensive program in Mathematics, Computer Science and Physics — 2022 – 2024

Profile

- Seeking a **research internship** ideally from early **July to late August 2026**.
- Second-year engineering student, curious and motivated, with a passion for solving complex problems.
- Actively involved in school associations and holds a valid driving license.

Programming Languages

Python, Rust, OCaml, C, C#, Bash, Java, Lua, SystemVerilog

Projects

Most of my project can be found on my GitHub .

Participation in the Autonomous Car Race at ENS Paris-Saclay 2025 (CoVAPSy)

High-level programming lead, INTech, the robotics association of Télécom SudParis

- Trained **PPO** agent in the Webots simulator using LiDAR + camera inputs with **PyTorch** + **Stable-Baselines3**
- Designed architectures with residual connections and two-layer convolutional blocks inspired by **ResNet** (Dropout + BatchNorm)
- Optimized models for embedded inference on Raspberry Pi 5 via **int8 quantization** and **ONNX** export
- Parallelized multiple simulation environments to accelerate training using IPC (named pipes) and process synchronization
- Project starred by lead developer of Stable-Baselines3

AI Training Sessions

Co-head of AI division, Télécom SudParis branch of Kryptosphère

- Designed and delivered ~10 training sessions (~1.5h each) combining slides and practical labs
- Covered a range of neural network architectures: MLPs, CNNs, RNNs, LSTM, GRU, ResNet
- Taught implementation and best practices in **PyTorch**, used regularly for 4+ years

Chess Project (TIPE) - Preparatory Class (MPI*)

- Created a custom chess language with explicit tokens for pieces, moves, actions, and board context
- Trained a small autoregressive RNN on translated chess datasets to study learned patterns
- Analyzed cosine similarity between tokens, revealing correlations between semantically related game elements
- Investigated RNN memory limitations for board state tracking (> 6-8 moves)

Digital Design Intern - NXP Semiconductors

- Analyzed a frequency estimation algorithm for carrier frequency offset (CFO) in wireless communications (Wi-Fi, Bluetooth, radio)
- Benchmarked against state-of-the-art estimators across different algorithm parameters and SNR scenarios
- Developed a testbench for an processing unit, enabling execution of custom assembly and implementated the algorithm
- Gained foundational experience in SystemVerilog and digital design

Swarm Rescue Challenge - Institut Polytechnique de Paris (Ongoing)

- Competing in a 2D multi-drone rescue challenge with 10 drones navigating mazes to save agents
- Implementing world representation and graph-based navigation using a **quad tree** to handle obstacles
- Fusing odometry and noisy GPS data with a **Kalman filter** for robust localization
- Developing pathfinding **(A* / D*)** and MLP-based trajectory following to maximize performance within the simulator

Campus Network Association (MiNET) — Télécom SudParis / IMT-BS

- Managed network and server infrastructure for 800 members (20+ Proxmox VE servers, HP ProLiant, VMs/LXC, Cisco devices)
- Deployed LLMs on servers using open-source services (vLLM, Ollama, llama.cpp, prima.cpp, gemma.cpp, Open-WebUI, LibreChat)

Learning the Rust Programming Language

- Implemented projects from scratch: JSON parser, PNG-inspired file format parser, Sudoku solver using backtracking
- Solved numerous Advent of Code 2024–2025 challenges, applying algorithmic problem-solving in Rust

INT Game Jam

- Developed a C-based rendering engine on Raylib entirely based on the ray-marching algorithm in a team of three
- Implemented maps as mathematical signed distance functions (SDFs) evolving over time and position in separate GLSL shaders