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 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 a strong team player.
- Holds a valid driving license.

Skills

Programming Languages

- Python
- Rust
- OCaml
- C
- C#
- Bash
- Java
- Lua
- System Verilog
- PHP

Frameworks

Machine Learning and Data Science: PyTorch, Stable-Baselines3, numpy, pandas, matplotlib Game Developpement: Unity, Garry’s Mod (Valve’s Source engine), s&box (Valve’s Source 2 engine) Micro-Controllers: Arduino, 8051 assembly for AT89 series Compilers: Java JFlex, Java CUP

Projects

Participation in the Autonomous Car Race at ENS Paris-Saclay

During my first year of engineering school, I joined INTech, the robotics and electronics association of Télécom SudParis, to represent my school at the ENS Paris-Saclay autonomous car race (CoVAPSy). As the high-level programming lead, I used the Webots simulator to train custom models capable of controlling an autonomous vehicle using data from a LiDAR sensor and an onboard camera. I implemented architectures featuring residual connections and two-layer convolutional residual blocks inspired by ResNet, along with training-enhancement techniques such as Dropout and Batch Normalization. To ensure efficient inference on embedded hardware (Raspberry Pi 5), I performed int8 quantization and exported the model to ONNX format. The training pipeline relied on the PPO algorithm. A key technical challenge was the parallelization of multiple simulation environments to accelerate data collection, which required robust inter-process communication via named pipes as well as reliable process synchronization mechanisms. The project has notably been starred by the lead developer of Stable-Baselines3 who published many times in A* conferences. The project is available on GitHub

AI Training Sessions

In 2025 I prepared about a dozen training modules of about 1 hour and a half each as co-head of the AI division of the Kryptosphère branch at Télécom SudParis and as the high-level programming lead of INTech, the robotics association. Each session included a presentation with slides and a practical lab so students could apply the concepts. The goal was to teach my classmates how different neural network architectures work such as MLPs, CNNs, RNNs, LSTM-RNNs, GRU-RNNs, and ResNets—and how to implement them using the PyTorch library, which I have been using regularly for over four years.

Applying language theory to Chess

During my second year of preparatory class I had to present a yearly project (TIPE) concerning games. I chose to apply the neural network knowledge I had to the case of the chess game. I wanted to create a new language to represent chess games that would be very explicit in terms of what is happening in the game an give a lot of contextual information. I would then train a small autoregressive model on a translated dataset of chess games to see what it retained from the training. The model learned how to move pieces on its own so it understood some of the rules. But it didn't have enough "memory" (I used RNNs) to keep track of the state of the board for more than about a dozen moves. Since my alphabet is very explicit with tokens corresponding to chess concepts (like a Rook, a Bishop, the action of eating, puting someone in check, the rows, the lines etc), the cosine similarity matrix showed some very interesting correlations between tokens of a similar type.

Internship at NXP Semiconductors

During my first year at telecom sudparis I did an internship at NXP semiconductors where I worked in a digital design team where I learned bases of System Verilog. The goal was to do analysis on a frequency estimation algorithm that would ideally be used in the future for estimating the carrier frequency offset (CFO) to enhance the quality of the communications in the case of wireless communications (wifi, bluetooth or radio). I had to compare it against to other state of the art estimators in performance and consumption for different parameters of the alogrithm and in different SNR scenarios. Then I had to help my internship tutor build a testbench around an internal processing unit to make it able to run our custom assembly and then implement the algorithm in that assembly to prove that it is feasible. In this internship I had to read and understand signal processing papers in order to implement the estimators described inside it.

Mandate in the Campus Network Association

I am currently serving in MiNET, the campus networking association for Télécom SudParis and IMT-BS. We ensure the infrastructure that provides internet access—both wired and Wi-Fi to 800 members. We also manage an infrastructure of more than 20 physical servers running Proxmox VE, hosting numerous VMs and LXC containers, along with networking equipment such as Cisco switches and WLC controllers. In this role I deployed an LLM on a server and tested many open-source solutions, including Ollama, llama.cpp, prima.cpp, gemma.cpp, Open-WebUI, and LibreChat.

Learning the Rust Programming Language

Since the summer holidays of 2025 I have been studying Rust. I enjoy it for its functional programming features and the security guaranties it offers over other languages. As part of my learning I built some projects from scratch such as a JSON parser, a parser for a PNG inspired file format, a Sudoku solver using backtracking, and solved many Advent of Code 2024 and 2025 problems.

Unity Video Game Project

I contributed to the development of a 3D-rendered video game. In a team of five, I created HLSL shaders, designed particle systems, planned the overall architecture (abstract classes, virtual classes, interfaces) for our C# project, and produced a UML diagram. I also programmed part of the boss behavior in C#.

INT Game Jam

I participated in a Game Jam where, in a team of three, we coded a rendering engine in C based entirely on the ray-marching algorithm. I programmed several maps that evolved over time and position, all implemented with GLSL shaders.