Pierre LIN

📱 (+33) 6 51 89 90 23 | ✉ pierre.lin.x@gmail.com | 🌐 Portfolio | GitHub | in LinkedIn

Looking for an **Internship** in **Software Engineering**.
Availability – **May 27th, 2024 / 3 Months**.

Technical Skills

- **Programming** – C | C++ | Java | Python | TypeScript.
- **Game Engines** – Unreal Engine | Godot.
- **Frameworks & Libraries** – Qt6 | SDL2 | OpenCV | ThreeJs.
- **Miscellaneous** – Git | SQL | CMake | LaTeX.

Education

Grenoble INP – Ensimag **2022 – 2025**
Master's Degree – Computer Science. Grenoble, France

- Relevant Courses : Algorithms • Data Structures • Object-Oriented Programming • Software Engineering • Web Development • Databases • Operating Systems • Concurrent Programming • Language Theory...

Sorbonne University **2020 – 2022**
Double Major – Mathematics & Computer Science. Paris, France

Personal Projects

Game Engine & Gameplay Programming (C++ / SDL2) ↗ **June 2023 – Current**

- Developed a game engine from scratch, including rendering, physics, collisions, UI, audio...
- Implemented an "entity component system" (ECS) software architecture to represent game world objects.
- Used Valgrind to solve memory leaks and GPerfTools to optimize the engine with code profiling.
- Created a game using the developed engine, including gameplay, world, entities, levels...

Survival Game (C++ / Unreal Engine) ↗ **February 2024**

- Built a full-fledged game using Unreal, covering gameplay, AI, UI, and audio.

Android Application (Java / Android Studio) ↗ **2019 – 2020**

- Created an Android mobile application featuring a music player, a book reader, and a mini-game.

Academic Projects

Software Engineering Project - Development of a Compiler ↗ **January 2024**
Java | Team of 5 | Grade : 18/20. Grenoble INP – Ensimag

- Development using the agile methodology with regular demos and progress updates to the client.
- Peer code review on GitLab and pair programming to produce high-quality code.
- Implemented the assembly code generation part of the compiler for two different architectures.
- Developed automated tests with Python to simplify the validation process of the compiler.
- Used Jacoco to do code coverage analysis, achieving a satisfying 96% coverage rate with our tests.

Implementation of the Fast Fourier Transform (FFT) ↗ **January 2022 – July 2022**
C | Team of 2 | Grade : 17/20. Sorbonne University

- Implemented an HPC (High Performance Computing) version of the FFT to multiply polynomials in $\mathbb{Z}/p\mathbb{Z}$.
- Optimized and analyzed code performance using techniques such as algorithmic, cache, compilation...
- Vectorized computation operations with AVX2, reducing the average execution time by 60%.

Languages

- **French** – Native | **English** – Advanced (TOEIC : 945/990) | **Chinese** – Native.