

Pierre LIN

☎ 06 51 89 90 23 | ✉ pierre.lin.x@gmail.com | 🌐 [Portfolio](#) | 🐙 [GitHub](#) | 💼 [LinkedIn](#)

Recherche un **Stage** dans le **Développement Logiciel**.

Disponibilité – **27 Mai 2024 / 3 Mois**.

Compétences

- **Programmation** – C | C++ | Java | Python | TypeScript.
- **Moteurs de Jeu** – Unreal Engine | Godot.
- **Frameworks & Bibliothèques** – Qt6 | SDL2 | OpenCV | ThreeJs.
- **Divers** – Git | SQL | CMake | LaTeX.

Formation

Grenoble INP – Ensimag

2022 – 2025

Diplôme d'Ingénieur – Ingénierie des Systèmes d'Information.

Grenoble, France

- Cours Pertinents : Algorithmique • Structures de Données • Programmation Orientée Objet • Génie Logiciel
• Bases de Données • Systèmes d'Exploitation • Programmation Concurrente • Théorie des Langages...

Sorbonne Université

2020 – 2022

Double Licence – Mathématiques & Informatique.

Paris, France

Projets Personnels

Moteur de Jeu & Gameplay Programming (C++ / SDL2) [🔗](#)

Juin 2023 – Présent

- Développement d'un moteur de jeu à partir de zéro, incluant le rendu, la physique, les collisions, l'UI, l'audio...
- Implémentation de l'architecture logicielle "entity component system" pour représenter les objets du moteur.
- Utilisation de Valgrind pour la résolution des problèmes mémoire et GPerfTools pour le profilage de code.
- Création d'un jeu avec le moteur développé, incluant le gameplay, le monde, les entités, les niveaux...

Jeu de Survie (C++ / Unreal Engine) [🔗](#)

Février 2024

- Développement d'un jeu complet sous Unreal avec implémentation du gameplay, de l'IA, de l'UI, et de l'audio.

Application Android (Java / Android Studio) [🔗](#)

2019 – 2020

- Développement d'une application Android incluant un lecteur de musique, un lecteur de livre, et un mini-jeu.

Projets Académiques

Projet Génie Logiciel – Développement d'un Compilateur [🔗](#)

Janvier 2024

Java | Équipe de 5 | Note : 18/20.

Grenoble INP – Ensimag

- Développement en mode agile avec des démonstrations et points d'avancement réguliers auprès du client.
- Revue de code par les pairs sur GitLab et programmation par pairs pour produire un code de qualité.
- Implémentation de la partie génération de code assembleur du compilateur pour deux architectures différentes.
- Développement de tests automatisés avec Python pour simplifier le processus de validation du compilateur.
- Analyse de couverture de code avec Jacoco, atteignant un taux de couverture de 96% avec nos tests.

Implémentation de la Fast Fourier Transform (FFT) [🔗](#)

Janvier 2022 – Juillet 2022

C | Équipe de 2 | Note : 17/20.

Sorbonne Université

- Implémentation HPC (High Performance Computing) de la FFT pour le produit de polynômes dans $\mathbb{Z}/p\mathbb{Z}$.
- Optimisation et analyse de la performance du code par des techniques d'algorithmique, cache, compilation...
- Vectorisation des opérations de calcul avec AVX2, réduisant le temps d'exécution moyen de 60%.

Langues

- **Français** – Natif | **Anglais** – Avancé (TOEIC : 945/990) | **Chinois** – Natif.