



GAMPAM

Tel. 06 76 33 02 36
contact@gampam.com

Laboratoire/Entreprise d'accueil
CNRS UMR6072 GREYC / GAMPAM
Contact

François Rioult <francois.rioult@unicaen.fr>
Paul Lenczner <contact@gampam.com>
Date limite de publication: 1 mars 2025

Stage IA générative pour conception de tableau de bord - Jeux video

Mots clés : IA générative, Jeux Web & Mobile

Type de contrat : Stage de master 2 en informatique

Durée : 4 à 6 mois

Gratification : 4.35€ / heure - 660€/mois

À propos de l'entreprise

GAMPAM est une entreprise spécialisée dans le développement de jeux web et mobiles. Parmi ses domaines d'intérêt figurent la création d'edugames, une catégorie de jeux dédiée à l'apprentissage et à la pédagogie, ainsi que la création de jeux en ligne.

Les motivations de GAMPAM sont de proposer un service no-code de développement assisté par IA pour concevoir les jeux, leurs mécaniques et leurs graphismes.

Mission

Le stage propose d'explorer, à l'aide d'un modèle de langue, la génération:

- **D'une documentation, aussi appelée GDD** (Game Design Document), lequel suit également des règles d'écriture strictes.

GDD: document de conception de jeu complet pour les templates de jeux à engendrer, qui détaille les mécaniques, les règles, les interfaces utilisateur et les spécificités techniques du jeu.

- **D'un tableau de bord à partir de notre langage maison BoxLang définissant la conception d'un jeu vidéo.**

BoxLang est un langage de style interne, inspiré de JSON, HTML et JavaScript, conçu pour créer des interfaces abstraites et adaptables. Il permet de construire des tableaux de bord personnalisables grâce à des fonctionnalités avancées comme la récursivité, l'itérabilité et le clonage. Ses modules facilitent la généralisation des interfaces graphiques, en les définissant de manière abstraite

Les récents développements en IA générative permettent l'exécution de prompts complexes pour la création de données structurées. L'ingénierie du prompt peut ainsi être abordée comme une recherche dans un espace vectoriel de programmes [1]. De plus, l'enrichissement du prompt par construction automatique d'une chaîne de pensée simule un raisonnement approfondissant les résultats usuels [2].

Pour spécialiser la base de connaissances du modèle, des techniques de RAG [3] permettent d'enrichir le prompt grâce aux résultats de l'interrogation de cette base. Des techniques de distillation sont également utiles [4] pour transférer les connaissances d'un grand modèle superviseur vers un modèle compact.



L'étudiant(e) réalisera une synthèse de l'état de l'art en matière de RAG et de distillation. Il/elle mettra en œuvre un processus de distillation pour obtenir un modèle compact capable de générer un tableau de bord à l'aide d'un langage maison, BoxLang, dédié à la création d'interfaces de contrôle.

Ce stage offre des opportunités de poursuite en thèse.

4. Contexte de Travail et Outils :

- Travailler sur du code destiné à être déployé dans un environnement distribué, en interaction avec divers serveurs et bases de données, pour une utilisation à grande échelle.
- Respecter les conventions de codage en JavaScript et mettre en place des mesures de sécurité contre la rétro-ingénierie, tout en réalisant des benchmarks réguliers et des tests unitaires.

5. Participation au Développement en Équipe, Environnement CI/CD :

- Intégrer une équipe internationale de développeurs seniors anglophones (Québec, Roumanie, Pologne, France).
- Participer activement aux réunions quotidiennes : un briefing le matin et un débriefing le soir.
- Collaborer via Discord et GitHub (branches, issues, tableaux Kanban) pour la synchronisation des tâches dans le cadre de notre gestion de projets Agile et la communication d'équipe.
- Utiliser des environnements d'intégration/déploiement continu.

Outils et Ressources :

Documentation Complète : Vous aurez accès à une documentation exhaustive sur BoxLang, notre moteur de jeu, ainsi que nos conventions d'écriture.

Extensions pour Éditeurs de Code : Nous fournissons des extensions spécifiques pour Vim et VSCode, facilitant votre intégration dans notre contexte de travail et le respect des conventions de codage.

Support de l'Équipe : Vous serez accompagné(e) par des développeurs seniors pour faciliter votre intégration et votre montée en compétences.

Profil recherché

- Étudiant de master 2 en informatique ou 3^è année d'école d'ingénieur.
- Maîtrise du JavaScript (vanilla),
- Bases solides en algorithmie,
- Intérêt pour l'abstraction, la généralisation et la conception de jeux,
- Bonne culture de l'IA, en particulier les LLM et leurs applications
- Bon niveau d'anglais (lu, écrit, parlé)

Phases de recrutement en deux étapes

- **Coding game en JavaScript** : Prévoir quelques heures pour réaliser le défi.

Accès au challenge:

<https://challenge.gampam.com/>

Identifiant : **unicaen**
Mot de passe : **stage**



Veillez envoyer votre code dans un **fichier .txt** à **contact@gampam.com**, en respectant les conventions de codage explicitement décrites dans la documentation (accessible via le point d'interrogation en haut à droite), accompagné de votre CV, relevés de notes, recommandation et lettre de motivation montrant l'intérêt pour la recherche dans ce domaine, avant le 10/03/2025.

Seul le niveau 1 (une étoile) est demandé ; cependant, la résolution des niveaux 2 et 3 sera grandement appréciée.

- **Suivi d'un entretien de 30 minutes** : Cet entretien portera sur des points théoriques que nous vous énoncerons à l'avance, afin que vous puissiez les explorer par vos propres recherches. Cet échange sera peut-être accompagné d'exercices pratiques supplémentaires.

Organisation du stage

Nous organisons deux réunions chaque jour : une le matin pour le briefing et une en fin de journée pour le débriefing. Nous restons disponibles tout au long de la journée pour répondre aux questions ou organiser des réunions supplémentaires si nécessaire.

Localisation : Télétravail avec possibilité de travail dans nos locaux à CAEN.

References

[1] Cholet, F. (2023). How I think about LLM prompt engineering, <https://fchollet.substack.com/p/how-i-think-about-llm-prompt-engineering>

[2] Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic Chain of Thought Prompting in Large Language Models. ArXiv, abs/2210.03493.

[3] Lewis, P. et al. (2020), Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, ArXiv, abs/2005.11401.

[4] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. ArXiv, abs/1503.02531.