

AI Game Generator – Projet R&D IA

Génération automatisée de Visual Novel jouable à partir d'un prompt utilisateur, par orchestration de modèles IA open-source.

Présentation

Ce projet explore la faisabilité d'un pipeline complet permettant à un utilisateur non-technique de générer un **visual novel (VN)** jouable à partir d'une **description textuelle simple**, sans écrire une seule ligne de code ni produire manuellement d'illustrations.

Le prototype combine plusieurs modules d'intelligence artificielle (LLM, diffusion) avec une orchestration logicielle (backend, frontend, export) afin de produire automatiquement :

- un **synopsis** et des **dialogues ramifiés**,
- des **illustrations contextuelles** générées par IA (arrière-plans),
- un **script RenPy** fonctionnel jouable localement.

Objectifs

- **Valider expérimentalement** un pipeline "idée vague → fiction jouable" à base d'IA.
- Proposer une **infrastructure open-source reproductible**, multi-plateforme et dockerisée.
- Réduire le coût cognitif de la création narrative interactive pour les néophytes.

Architecture du pipeline

```
Prompt utilisateur
  |
  ▼
[LLM local via Ollama] → Synopsis + dialogues + choix
  |
  ▼
[Script EMF-CLI] → Génération d'images (800x600)
  |
  ▼
[Backend Django + UI] → Orchestration, prévisualisation
  |
  ▼
[Export RenPy] → Compilation automatique des scènes en .rpy
```

PROF

Docker & Multiplateforme

L'ensemble du système a été encapsulé dans une infrastructure **Dockerisée** permettant :

- une exécution homogène sous **Linux, macOS, Windows** (x86 ou ARM),
- une **réplication simple** des tests d'inférence, y compris en environnement contraint,

- un lancement **one-click** pour les utilisateurs finaux.



Démarche R&D

Ce projet s'inscrit dans une logique de **recherche appliquée**, articulée autour de plusieurs axes :

- étude des **capacités narratives des LLM** en environnement local,
- évaluation de la **cohérence multimodale texte/image**,
- génération de scripts **.rpy** sans règles codées manuellement (prompt → structure code).

L'évaluation des performances repose sur des métriques objectives (temps d'inférence, usage mémoire) et sur une validation humaine qualitative (cohérence narrative, jouabilité, qualité des visuels).



Utilisation du projet

Ce projet a été conçu pour être lancé très facilement via Docker. Aucune installation complexe n'est requise.



Prérequis

- [Docker](#) installé sur votre machine
- [Docker Compose](#) (inclus dans Docker Desktop)



Lancer le projet

Une fois le projet récupéré depuis le depo git, lancez un terminal et exécutez la commande suivante :

```
# Construire les images et lancer les services
docker-compose build && docker-compose up -d
```



Accéder à l'interface

Une fois le lancement terminé, ouvrez votre navigateur à l'adresse suivante :

```
http://localhost:8080/
```

Vous pouvez désormais générer votre propre visual novel directement depuis l'interface !



Équipe projet

Nom	Rôle
Kalim Moussa	Chef de projet & IA
Mohamed Driss Bendriss	Intégrateur Full-stack & Docker

Nom	Rôle
Marlind Tahiri	RenPy & Export
Assalas Lakrouz	RenPy & Pipeline <code>.rpy</code>
Bryan Dam	Backend
Divino Schaeffer	Backend
Houssam El Bouhi	Frontend UI
Mohamed Atmimou	Frontend dialogues



Licence

Ce projet est distribué sous licence **MIT**, sauf mention contraire dans les sous-modules externes. Les modèles utilisés restent soumis aux conditions de leurs auteurs respectifs (Ollama, Hugging Face, etc.).

Voici la section à ajouter à la fin du README :