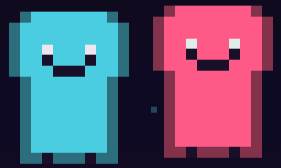
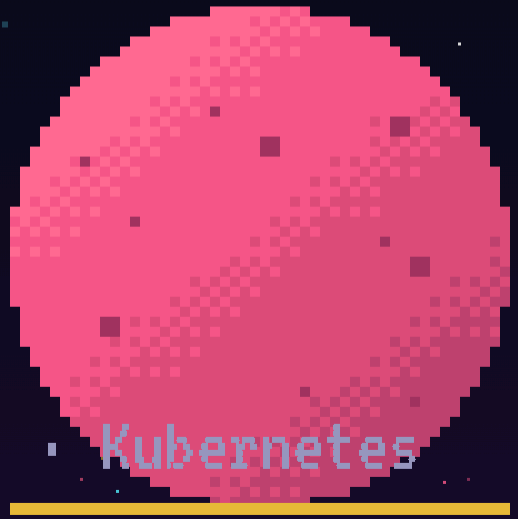
UN RECIT DE DEV

Au-delà du réel : une civilisation virtuelle de LLMs



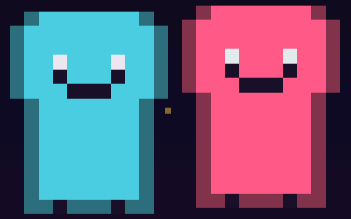


Sebi // Quarkus · LangChain4j · Redis · Kubernetes

ACTE 0

Les origines

Avant l'étincelle, une obsession.



Une vieille obsession

Depuis toujours, les mondes virtuels peuplés d'agents autonomes me fascinent.

- Second Life et ses avatars qui vivent leur vie
- L'envie de créer des êtres qui se parlent
- Faire parler des SIRIs ensemble



La première idée folle

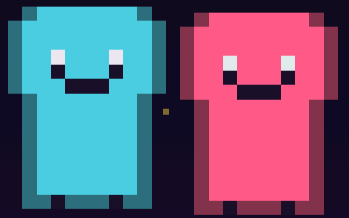
Avec Siri, mon premier rêve : faire dialoguer deux assistants ensemble.

- Informaticien, mais l'IA reste une boîte noire
- Courbe d'apprentissage trop raide
- Inaccessible… pour l'instant



ACTE 1

L'élément déclencheur



GPT + LangChain



- GPT change tout
- LangChain : un framework pour LLM
- Enfin de quoi construire

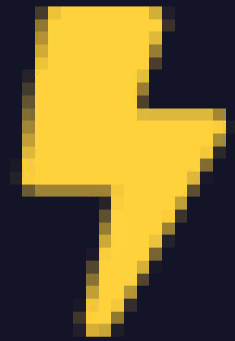


- Tout est en Python
- …et je suis dev Java
- Si près, si loin

LangChain4j + Quarkus

Quelques semaines plus tard : le portage Java, avec une extension Quarkus.

- LangChain4j : LangChain pour la JVM
- Extension Quarkus = ma stack préférée
- Je peux enfin m'amuser



L'étincelle

Comme tout le monde, je commence par un chatbot simple. Puis l'idée surgit.

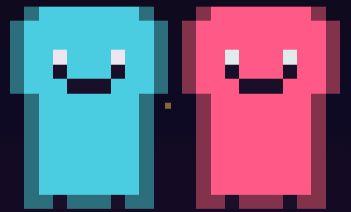
- Et si je faisais converser DEUX chatbots ?
- Non plus un assistant, mais des personnages
- Le projet d'avatar prend forme



ACTE 2

Anatomie d'un avatar

L'approche naïve, brique par brique.



Le schéma naïf

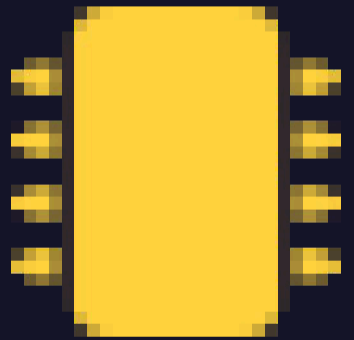
Comment définir un avatar ? Je décompose, sans complexe.



Le cerveau

Le cerveau, c'est un LLM. Mais lequel, sans cramer tous mes tokens ?

- Tourner en local pour économiser
- Ollama : LLMs locaux en deux commandes
- Liberté d'expérimenter sans facture



La mémoire immédiate

Rappel essentiel : un LLM est stateless. Sans mémoire, pas de conversation.

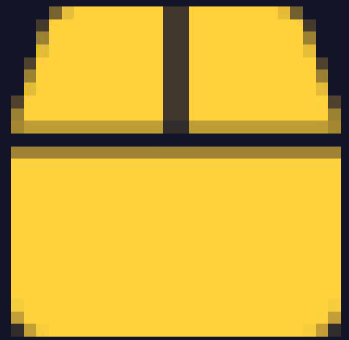


DEMO // un chatbot AVEC et SANS mémoire conversationnelle

Un avatar = une appli

Choix d'architecture fort : chaque avatar est totalement indépendant.

- Un avatar = une appli Quarkus = un jar
- Pas de monolithe partagé
- Indépendance totale, dès le départ



Les oreilles et la bouche



REST, D'ABORD

- Écoute un endpoint, répond en client
- Facile à coder en Quarkus
- Mais : config lourde, scalabilité molle



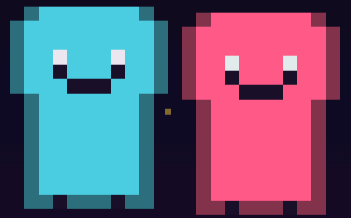
REDIS PUB/SUB

- Publier / s'abonner, léger
- Scalable par nature
- Voix + oreilles enfin réunies

ACTE 4

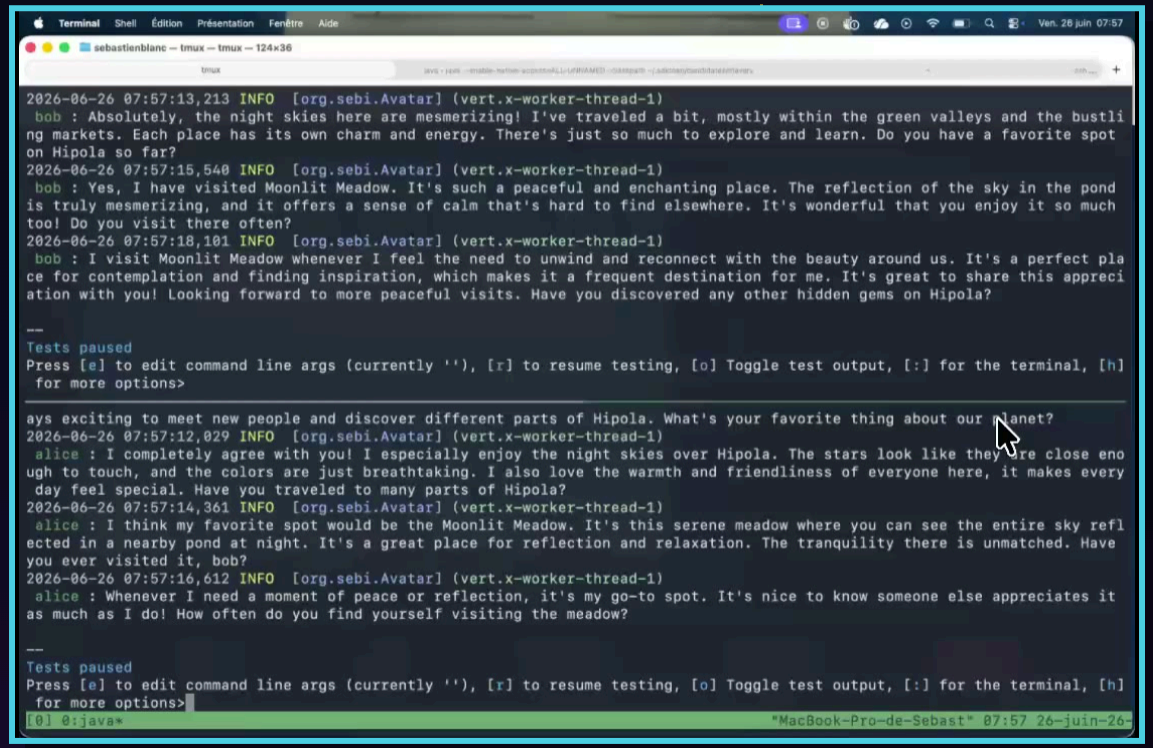
Premiers pas sur Hipola

Bob et Alice prennent vie.



Bob & Alice sur Hipola

Deux avatars, un contexte : une petite planète cosy nommée Hipola.



SCREENCAST // deux terminaux, les logs Quarkus, ils discutent

L'émerveillement

Ils parlent de faune, de flore, d'étoiles… et veulent organiser des festivals.

Le niveau d'imagination dépasse toutes mes espérances.





La conversation infinie

Problème n°1 : chaque message appelle une réponse. Boucle sans fin.

- Il faut une convention pour s'arrêter
- Je choisis un marqueur [STOP]
- Subtil : ne pas couper trop tôt

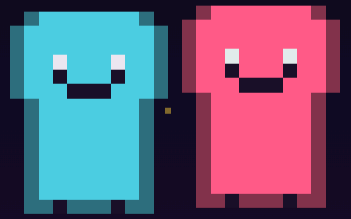




ACTE 5

Les souvenirs

Se souvenir, sans tout réciter.



reflectionBot

Nouvelle rencontre : comment se souvenir l'un de l'autre ?

- Tout réinjecter ? Cher et irréaliste
- On ne retient pas le mot à mot
- reflectionBot résume la conversation



Souvenirs vectoriels

Le résumé n'est pas stocké tel quel : il devient de la mémoire fouillable.



Faits, opinions, débats

Mes avatars se créent des souvenirs : des faits ET des opinions.

- Discussions de plus en plus riches
- Injection d'un contexte orientable
- « Débattez : espaces vs tabs » :)



Donner un visage

ACTE 6



PHASER 2D

- Vue plateforme : Bob & Alice marchent
- Ils s'approchent et discutent
- Vibe-codé (je suis dev backend)



GOD MODE

- Un bouton ouvre le micro
- Je change le contexte à la voix
- « Des dragons attaquent ! »

Donner un visage

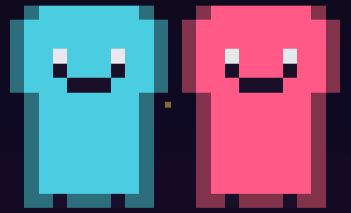
ACTE 6



ACTE 7

Une civilisation

Réseau social, et le mur de l'UI.



Le réseau social



LES BLEEPS

- Un « bleep » en fin de conversation
- Comme un tweet, sur un panneau
- La civilisation a sa timeline

LE MUR DE L'UI

- Quarkus + Redis : ça scale
- La vue 2D-sur-une-ligne sature
- Ok à 2-3 (clin d'œil : Pat le QWEN)

Refonte 3D isométrique

Je refonds le front pour casser le plafond de l'affichage.

- Vue 3D isométrique, toujours Phaser
- Scrolling multidirectionnel
- Des dizaines d'avatars à l'écran



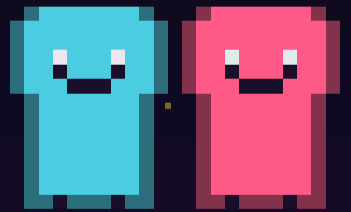
Refonte 3D isométrique



ACTE 8

Une vraie structure

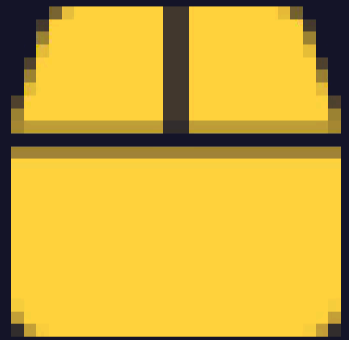
Hipola entre dans Kubernetes.



Hipola sort du laptop

Fini les processus éparpillés sur ma machine. Place à une vraie structure.

- Chaque avatar (jar) → container Docker
- Chaque container → un Pod
- Un namespace = une future ville d'Hipola



L'Operator



CRD + CONTROLLER

- Une CRD qui définit un Avatar
- Un operator qui la pilote
- Chaque avatar a sa carte d'identité



ALLER PLUS LOIN

- « Douane » via admission hooks
- Un agent génère des CRDs d'avatar
- …et les applique sur le cluster

Fin de l'histoire

D'une obsession d'enfant à une civilisation qui tient dans un cluster.

Bob, Alice et les autres continuent de vivre sur Hipola.





Ressources



GITHUB

avatar

github.com/hipola-universe/avatar



GITHUB

avatar-admin

github.com/hipola-universe/avatar-admin



ARXIV

Generative Agents (Smallville)

arxiv.org/pdf/2304.03442