

# Réveillez-vous, Développeur.

Le monde du logiciel a changé.



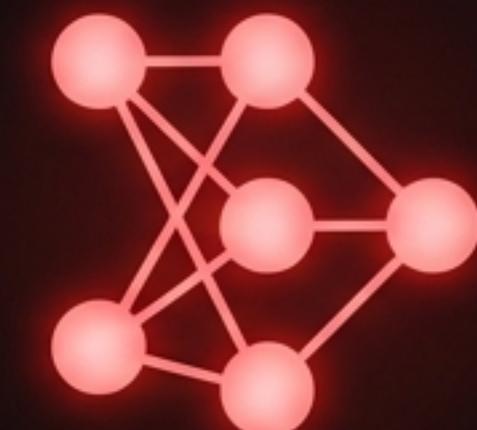
## LA PILULE BLEUE



## LA PILULE BLEUE

Vous restez un programmeur.  
Vous écrivez du code, ligne par ligne.  
Le système est votre outil.

## LA PILULE ROUGE

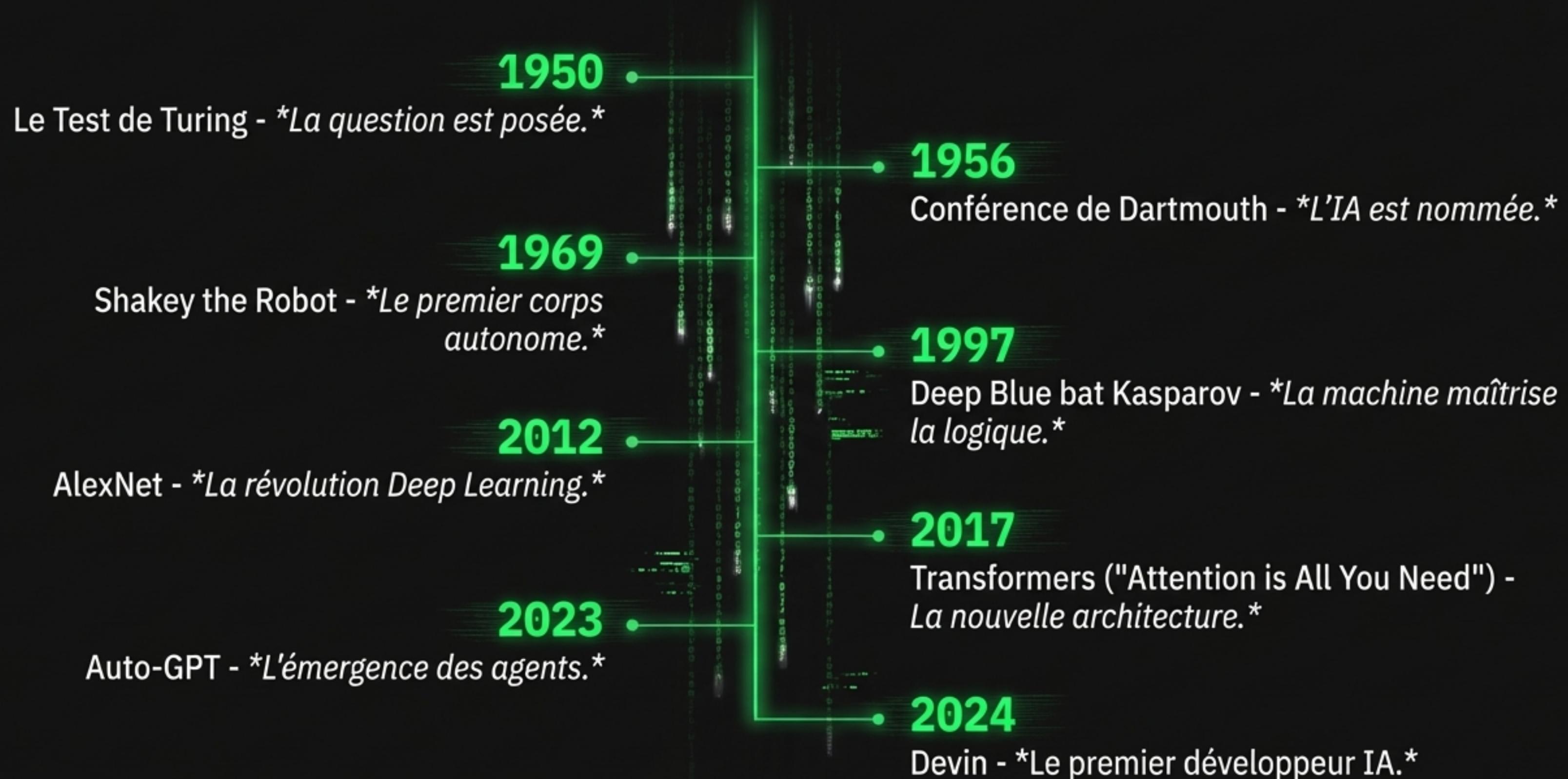


## LA PILULE ROUGE

Vous devenez un architecte.  
Vous dialoguez avec des systèmes autonomes.  
Vous êtes le chef d'orchestre.

# Quelle réalité choisissez-vous ?

# La Genèse de la Matrice : Une Histoire Accélérée



# Qu'est-ce qu'un Agent IA ?



Un **agent IA** est un système d'intelligence artificielle capable de **percevoir** son environnement, de **raisonner** sur ses objectifs, de **planifier** des actions, et **d'exécuter** ces actions de manière **autonome** pour atteindre ses buts.

Ce n'est plus un outil. C'est un collaborateur.

# ANATOMIE D'UN AGENT MODERNE

**LA MÉMOIRE : CONTEXTE & CONNAISSANCES**  
**TYPES :** COURT TERME (FENÊTRE DE CONTEXTE), LONG TERME (BASE VECTORIELLE / RAG).

**LE CERVEAU : LARGE LANGUAGE MODEL (LLM)**

**FONCTIONS :** COMPRÉHENSION, RAISONNEMENT, GÉNÉRATION.  
**EXEMPLES :** GPT-4o, CLAUDE 3.5, GEMINI.

**LES OUTILS : CAPACITÉS ÉTENDUES**

**PRINCIPE :** "FUNCTION CALLING".  
**EXEMPLES :** ACCÈS WEB, EXÉCUTION DE CODE, APIs EXTERNES, CONTRÔLE SYSTÈME.

**LE PLANIFICATEUR : STRATÉGIE D'ACTION**

**FONCTIONS :** DÉCOMPOSITION DE TÂCHES, EXÉCUTION SÉQUENTIELLE, ADAPTATION.

# Function Calling : Comment l'Agent Agit sur le Monde



"Quel temps fait-il à Paris ?"

## Requête Utilisateur

## Analyse du LLM

Le LLM détermine qu'un outil est nécessaire.

## Génération de l'Appel

```
{ "tool": "get_weather",  
  "parameters": {  
    "city": "Paris"  
  } }
```

## Exécution par le Système

Le système appelle la fonction météo réelle.

## Intégration de la Réponse

"Il fait 22°C à Paris."

# Il n'y a pas de cuillère.



Le code n'est plus la seule réalité.  
La nouvelle interface, c'est le langage.

Le **Prompting** est l'art de plier cette nouvelle réalité à votre volonté.

# Apprendre le Kung Fu : Les Fondamentaux du Prompt

## Mauvais Prompt

Vague et Imprécis

Écris du code pour un robot.

## Bon Prompt

Spécifique, Structuré, Contextualisé

Tu es un expert en robotique utilisant ROS 2 Humble.

**Tâche** : Écris un nœud d'éditeur Python simple qui publie un message "Hello World" sur le topic "/chatter".

**Contraintes** :

- Utilise la bibliothèque `rclpy`.
- Le message doit être de type `'std_msgs/String'`.
- Publie le message à une fréquence de 1 Hz.

**Format de sortie** : Fournis le code complet dans un seul bloc formaté en Python.

# Le Niveau Supérieur : La Chaîne de Pensée (Chain-of-Thought)

Ne demandez pas seulement la réponse. Demandez le **raisonnement**.

Problème : Un robot à 0.5 m/s doit parcourir 10m, tourner de 90° (2s), puis parcourir 5m. Temps total ? Résous ce problème étape par étape.

Réponse de l'Agent:

Raisonnement :

1. D'abord, calcule le temps pour parcourir 10 mètres.  
 $\text{Temps1} = \text{Distance} / \text{Vitesse} = 10\text{m} / 0.5 \text{ m/s} = 20\text{s}.$
2. Ensuite, ajoute le temps de rotation.  
 $\text{TempsRotation} = 2\text{s}.$
3. Puis, calcule le temps pour parcourir 5 mètres.  
 $\text{Temps2} = 5\text{m} / 0.5 \text{ m/s} = 10\text{s}.$
4. Enfin, additionne tous les temps.  
 $\text{TempsTotal} = 20\text{s} + 2\text{s} + 10\text{s} = 32\text{s}.$

Réponse finale : Le trajet total prendra 32 secondes.

# J'ai besoin d'armes. Beaucoup d'armes.

Votre arsenal pour construire le futur : l'écosystème Google AI.

## Gemini

### LE CERVEAU

Le modèle de langage multimodal au cœur de l'opération.



## NotebookLM

### LE BIBLIOTHÉCAIRE

L'assistant de recherche ancré dans vos documents pour une mémoire parfaite.



## Antigravity

### L'OUVRIER

La plateforme de développement "agent-first" où vos idées prennent vie.

# Focus sur le Cerveau : Gemini 3 Pro



TEXTE



IMAGE



VIDÉO



AUDIO



## 🧠 Thinking Mode

Génère une chaîne de pensée interne *avant* de répondre. Réduit les hallucinations de 60%.



## ⌚ Multimodalité Native

Comprend simultanément le texte, les images, la vidéo et l'audio.



## 📚 Contexte 1M+ Tokens

Capable d'analyser des bases de code ou des documentations entières en une seule fois.



## 🔧 Tool Use Natif

Sait instinctivement quand et comment utiliser des outils externes.

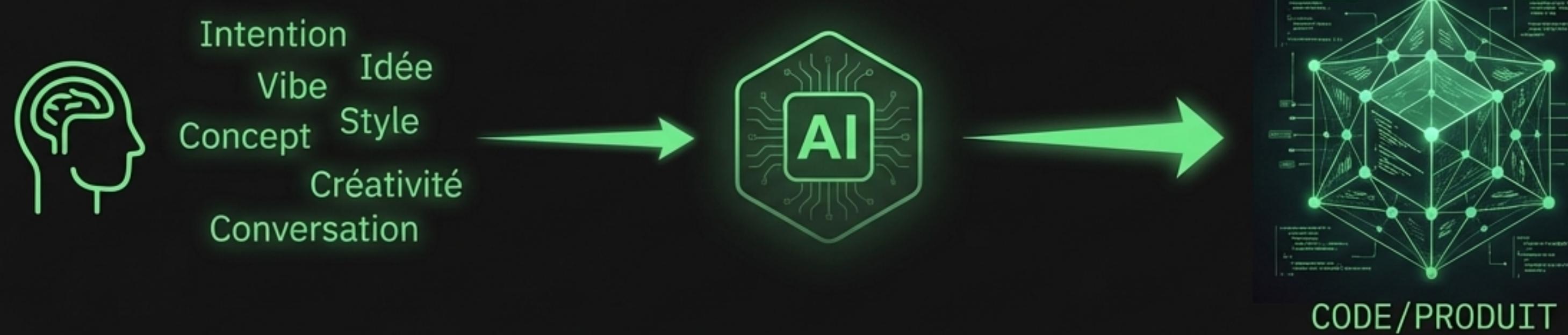
# Antigravity : Bienvenue dans le "Construct"

“**Antigravity** n'est pas juste un IDE avec de l'IA. C'est un environnement où les **agents autonomes** prennent en charge des **tâches complexes de bout en bout**.” – Blog Google Developers, 2025



Aspect	IDE Traditionnel (assisté)	Antigravity (agentique)
Paradigme	Autocomplétion	Agent autonome
Scope	Ligne / Fonction	Projet entier
Planification	Manuelle	Automatisée (multi-étapes)
Votre Rôle	Écrire du code	Guider l'agent

# La Nouvelle Philosophie : Le "Vibe Coding"



**Définition:** Vous décrivez **l'intention**, la **"vibe"**, ce que vous voulez.  
**L'IA code.** Vous itérez par la **conversation**.

VOUS	L'IA
Vision	Vélocité
Créativité	Connaissance Technique
Jugement	Exécution

# Le Programme de Saut : C'est l'heure de voler

**Objectif de la Session:** Créer une application web complète en 3 heures en collaborant avec un agent IA.

## Organisation:

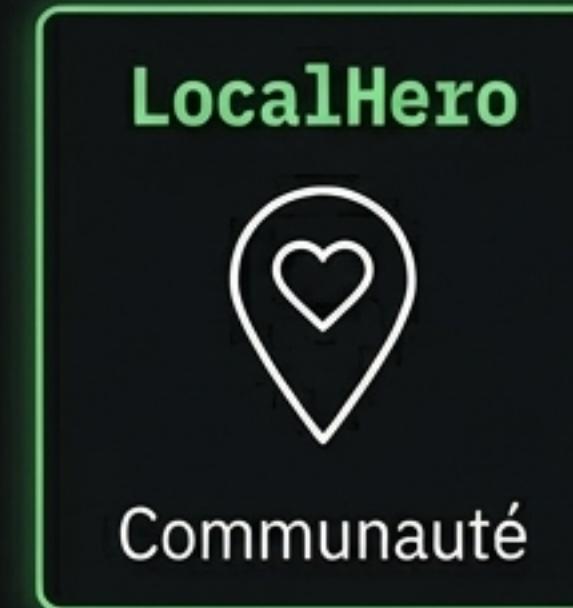
4-5 Équipes

4 Rôles en rotation :

- **Pilote IA:** Formule les prompts.
- **Co-pilote:** Suggère des améliorations.
- **Testeur:** Identifie les bugs en continu.
- **Documenteur:** Note les prompts efficaces.



# La Matrice des Projets : Choisissez votre mission



# Les Règles de l'Engagement

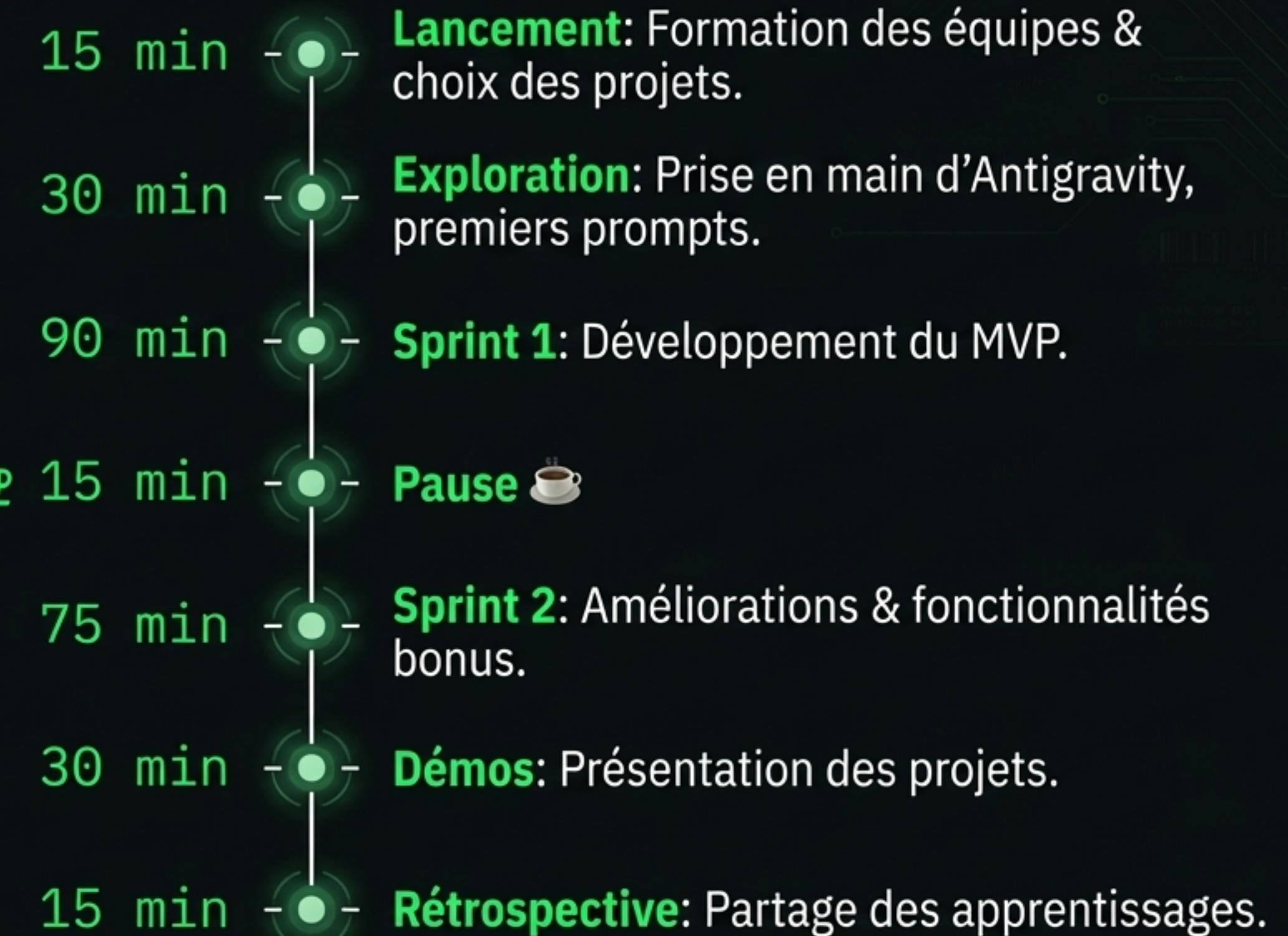
## Conseils pour réussir

- **Commencez simple** : Un MVP, puis itérez.
- **Soyez ultra-spécifique** : "Un bouton bleu en haut à droite" > "Améliore l'UI".
- **Testez en continu** : Après chaque changement majeur.

## Quand l'Agent se trompe ("Dodge this!")

1. **Reformulez**: Essayez une autre façon de demander.
2. **Décomposez**: Demandez une seule chose à la fois.
3. **Donnez du contexte**: "Dans le fichier `app.js`, dans la fonction `renderButton`..."
4. **Montrez un exemple**: "Je veux un résultat qui ressemble à [snippet de code]".

# Planning de la Session



# La Transformation : Du Code à l'Architecture

Avant

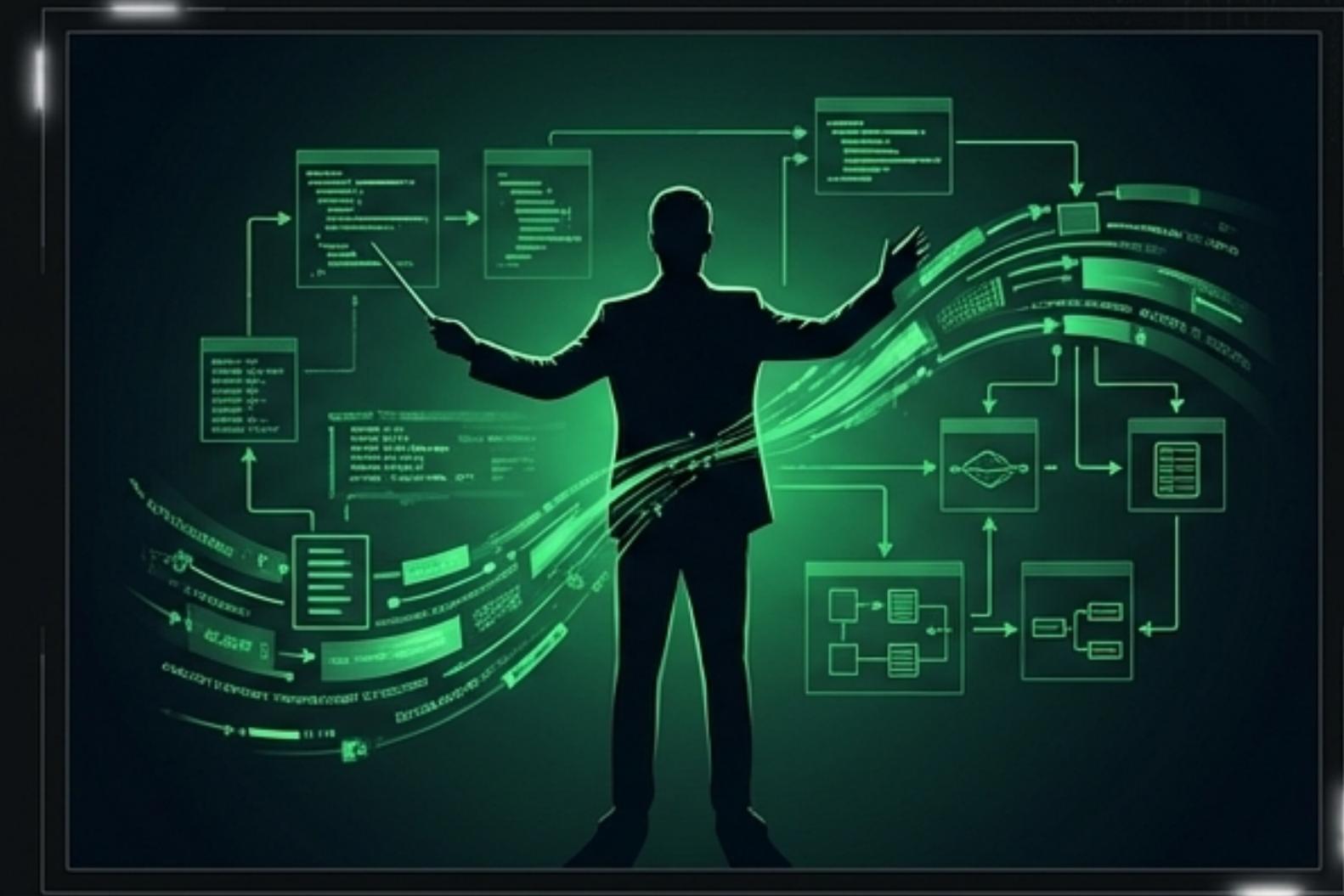


Focus : La syntaxe, l'algorithme.

Unité de travail : La ligne de code.

Question : "Comment vais-je coder ça ?"

Après



Focus : L'objectif, le système.

Unité de travail : L'intention.

Question : "Comment vais-je guider l'agent pour qu'il construise ça ?"

> Le meilleur code est celui  
que vous n'avez pas à écrire.

> - Jeff Atwood

**Le système est à vous maintenant.  
Construisez le futur.**