

# **COURS INTRODUCTION À IHM**

# COURS INTRODUCTION À IHM

BUT2 Informatique

## **De la ligne de commande aux interfaces conversationnelles**

*Pour développeurs et développeuses  
curieux·ses*

Loin d'être un simple sujet de design, la conception d'interfaces est au cœur du métier de développeur. Que vous créiez une application mobile, un site web ou un outil interne, l'interface est le seul point de contact entre votre code et l'utilisateur final. Comprendre son histoire et ses principes fondamentaux, c'est se donner les moyens de construire des produits qui ne sont pas seulement fonctionnels, mais aussi adoptés, appréciés et efficaces.

# **Plan de la Partie 1**

1. L'UI, ça compte ! (Principes fondamentaux)
2. La Révolution Graphique (Années 70-80)
3. L'Ère du Desktop & du Web (Années 90)
4. Le Big Bang Mobile (Années 2000 ->)
5. La Nouvelle Frontière : L'IA

Ce plan nous fera voyager à travers le temps. Nous commencerons par établir pourquoi l'UI/UX est un enjeu business et technique crucial. Ensuite, nous revivrons les moments clés de l'histoire des IHM : la naissance de l'interface graphique, sa démocratisation sur les ordinateurs de bureau, l'explosion du web, la révolution du mobile, pour enfin arriver à la dernière grande transformation en cours : l'intégration de l'intelligence artificielle.

**Pourquoi ce cours ?**

**Objectif :** Comprendre que l'UI n'est pas juste du "cosmétique".

C'est un **moteur** d'adoption, d'efficacité et de satisfaction. Une bonne IHM peut transformer un produit complexe en outil puissant et intuitif.

**Vous n'êtes pas des designers, mais des développeur·es conscient·es de l'impact de votre travail sur l'utilisateur final.**

En tant que développeur·se, une mauvaise UI a aussi un coût technique :

- **Code plus complexe :** Des contournements (workarounds) sont souvent nécessaires pour pallier des défauts de conception.
- **Maintenance difficile :** Un workflow illogique pour l'utilisateur se traduit souvent par une machine à états ( state machine ) complexe et difficile à maintenir dans le code.
- **Baisse de la qualité :** Des bugs sont souvent signalés alors qu'il s'agit en réalité de problèmes d'utilisabilité.

# Définitions Clés

**UI (User Interface / Interface Utilisateur) :** Le "comment". C'est la partie visible et tangible de l'application avec laquelle l'utilisateur interagit.

Exemples : Boutons, menus, formulaires, agencement visuel, typographie.

**UX (User Experience / Expérience Utilisateur) :** Le "ressenti". C'est l'expérience globale, le voyage émotionnel et pratique de l'utilisateur avec le produit.

Exemples : Est-ce facile de s'inscrire ? Est-ce que l'application est rapide ? Est-ce que je me sens en confiance en entrant mes données ?

**L'UI est une composante cruciale de l'UX.** Une bonne UI contribue à une bonne UX, mais une bonne UX nécessite plus qu'une simple belle UI (performances, support, etc.).

# **Le Coût d'une Mauvaise UI**

**Frustration utilisateur** -> Abandon de l'application, mauvais avis.

**Erreurs utilisateur** -> Perte de données, actions incorrectes (ex: commander le mauvais produit).

**Temps de formation rallongé** -> Coût élevé pour les logiciels d'entreprise.

**Charge cognitive élevée** -> L'utilisateur doit réfléchir pour utiliser votre produit au lieu de se concentrer sur sa tâche.

**Support client surchargé** -> Des appels pour des questions qui auraient dû être résolues par l'interface.

**Image de marque dégradée** -> "Le logiciel de cette entreprise est inutilisable."

**Un impact direct et mesurable sur le chiffre d'affaires et la réputation.**

## **Exemple #1 - La Mauvaise UI**

## **Une interface illogique, complexe ou trompeuse :**

- Workflow non intuitif (ex: devoir aller dans "Paramètres avancés" pour une action courante).
- Hiérarchie visuelle absente (tout semble avoir la même importance).
- Feedback utilisateur nul (ai-je bien cliqué ? l'action est-elle en cours ?).
- **Dark Patterns** : Des designs trompeurs qui poussent l'utilisateur à faire des actions qu'il ne souhaite pas (ex: un bouton "Accepter" très visible et un "Refuser" caché).



A



B

Exemple physique : "Comment obtenir simplement un café ?"



Exemples

## **Exemple #2 - La Bonne UI**

## **Une interface qui guide et se fait oublier :**

- **Hiérarchie claire** : Les titres sont plus gros, les actions principales sont mises en avant.
- **Affordance** : Les éléments suggèrent leur propre fonction (un bouton "ressemble" à un bouton, un texte souligné "ressemble" à un lien).
- **Feedback immédiat** : Un bouton change d'état au clic, une barre de chargement apparaît, un message de succès s'affiche.
- **Cohérence** : Les mêmes icônes et actions se retrouvent aux mêmes endroits dans toute l'application.

**Traveling to Switzerland**  
 11 Nov - 16 Nov • 11:00 AM

**Have a Good day, Wendy** 🙌

Fuel your days with the boundless enthusiasm of a lifelong explorer.

I want to... 📅 ★ ▶

🕒 Now 🕒 Tomorrow 🕒 Next week 🕒 Custom

Upcoming Schedule

December 2023

	Sun 10	Mon 11	Tue 12	Wed 13	Thu 14
11 am	Imperial Dimsum 🕒 10 Dec			Explore Forest Park 🕒 13 Dec • 🕒 10:00 AM	
12 am		Camping at Ranca Upas 🕒 11 Dec - 12 Dec • 🕒 11:00 AM			Mini Soccer 🕒 14 Dec

**Jane Cooper** 15:00  
• Online

**Jenny Wilson** 13:45  
• Online

**Brooklyn Simon** 10:20  
• 10 Minutes ago

**Theresa Angel** 08:30  
• 30 Minutes ago

**Kim Minji** Monday  
• 8 Minutes ago

**Brian Tracy** Sunday  
• 20 Minutes ago

Morning 🌞 12:47

Lets join us Wendy! 12:50

13:00 Sure Jenny 😊

**Sunset at Jimbaran**  
🕒 16 Nov • 🕒 16:00 PM

Reject Accept

13:20 That's cool, see you soon 🌞

Enter Text... ▶

# **Les Lois de l'UX**

## Loi de Hick :

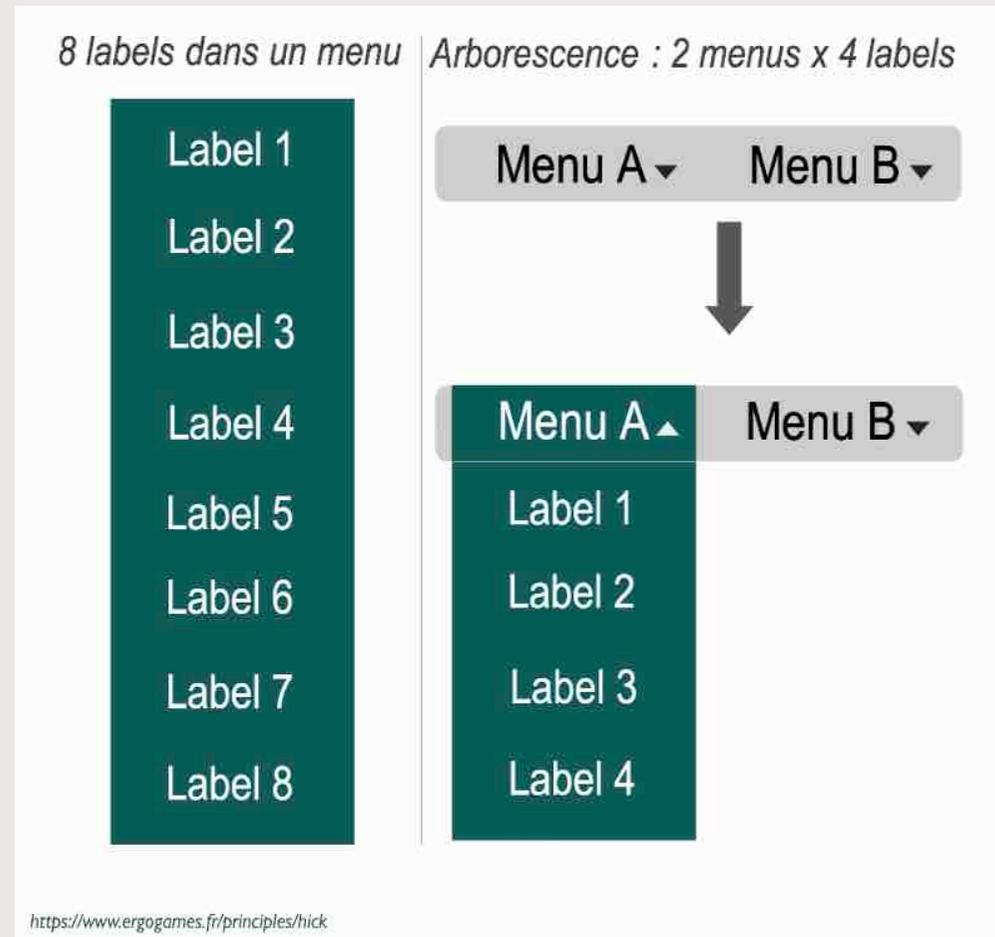
"Le temps pour prendre une décision augmente avec le nombre et la complexité des choix."

## Application concrète :

Au lieu de présenter 10 options dans un menu, regroupez-les en catégories logiques. Dans un formulaire, privilégiez un bouton "S'inscrire avec Google" et un lien "S'inscrire avec une email" plutôt qu'une liste de 15 fournisseurs de connexion.

**Simplifiez et guidez. Moins il y a de choix, plus la décision est rapide.**

## [Exemple Loi de Hick : formulaire simple vs complexe]



Un menu simple et un menu surchargé.

## Loi de Fitts :

"Le temps pour atteindre une cible est une fonction de la distance à la cible et de la taille de la cible."

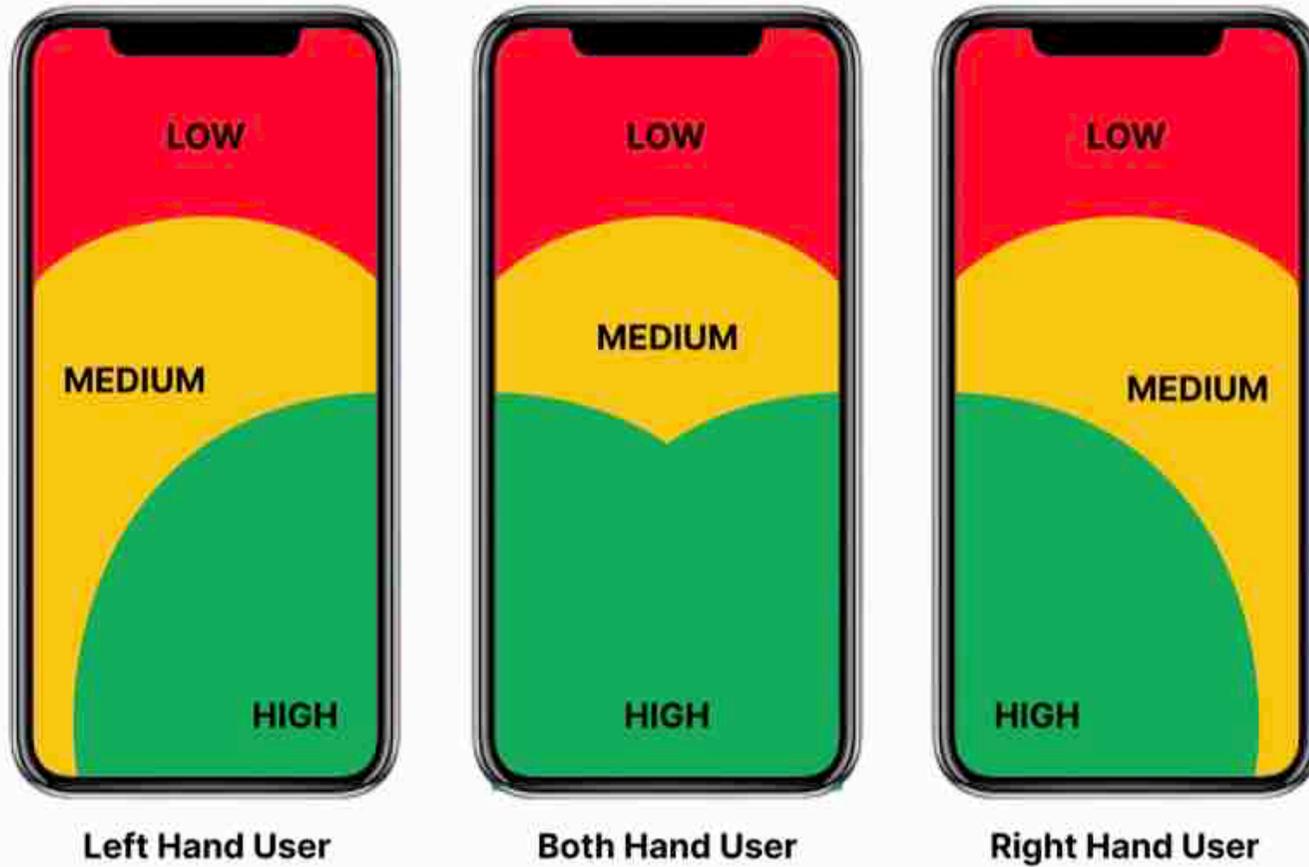
## Application concrète :

- Sur mobile, la barre de navigation avec les actions principales (Accueil, Recherche, Poster) est en bas, à portée de pouce.
- Les boutons de confirmation ("Acheter", "Envoyer") doivent être plus grands et plus faciles à cliquer que les boutons d'annulation.
- Les 4 coins d'un écran sont les zones les plus rapides à atteindre avec une souris.

**Les cibles importantes doivent être grandes et faciles d'accès.**

[Fitts Law]

### FITTS'S LAW



## Loi de Jakob :

"Les utilisateurs passent la plupart de leur temps sur d'autres sites/applications. Cela signifie qu'ils préfèrent que votre site fonctionne de la même manière que tous les autres sites qu'ils connaissent déjà."

## Application concrète :

Ne réinventez pas les icônes universelles. Un panier d'achat est un panier, le logo est en haut à gauche et ramène à l'accueil, un lien est souligné ou coloré. Enfreindre ces conventions force l'utilisateur à apprendre un nouveau langage, ce qu'il fera rarement.

**Respectez les conventions.  
L'innovation doit servir  
l'utilisateur, pas votre ego.**

**Avant l'UI Graphique**  
**L'ère du CLI (Command Line Interface)**

- **Interaction** : Purement textuelle. L'utilisateur doit connaître la syntaxe exacte des commandes.
- **Contexte** : Née à une époque où les ressources (CPU, mémoire, affichage) étaient extrêmement limitées. C'était la forme d'interaction la plus efficace pour la machine.
- **Avantages** : Puissance, précision, scriptabilité, automatisation. C'est pourquoi elle est encore massivement utilisée par les développeurs ( git , docker , ssh , npm ...).
- **Inconvénients** : Courbe d'apprentissage très raide, aucune "découvrabilité" (on ne peut pas deviner les commandes).

**Le monde appartenait à ceux qui connaissaient le manuel par cœur.**

# **Exemple CLI**

## **Un terminal Unix/Linux.**

La commande `ls -la` liste les fichiers du répertoire courant avec leurs détails. C'est extrêmement puissant pour un expert, mais totalement opaque pour un novice. Il n'y a aucun indice visuel sur ce qui est possible.



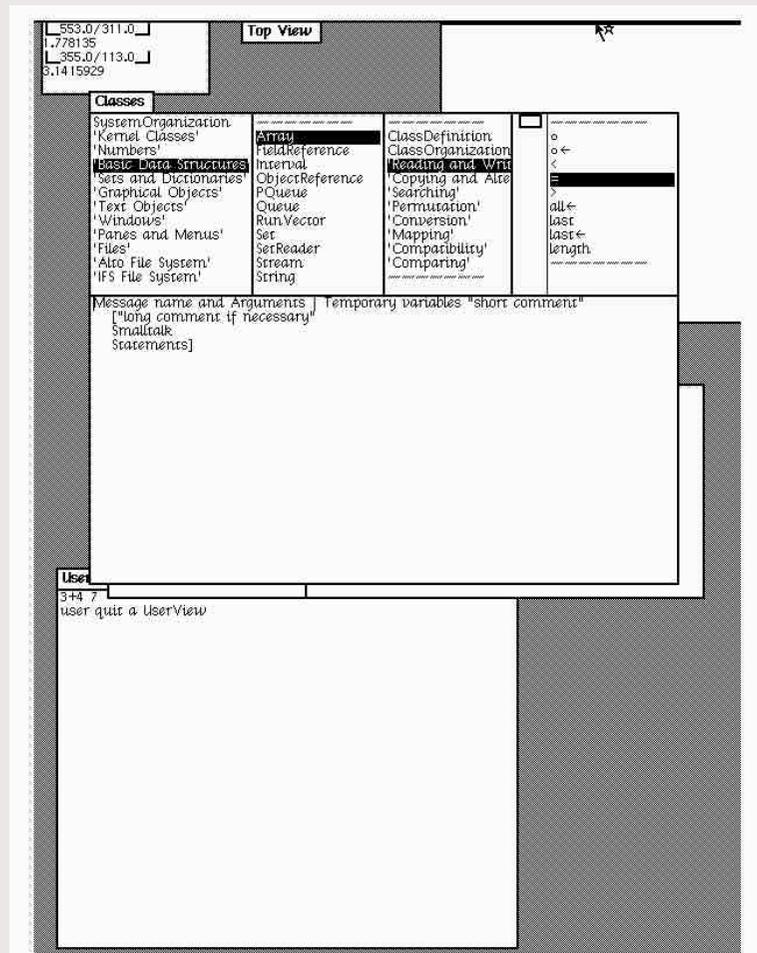
**La Révolution : Xerox PARC (1973)**

**Le Xerox Alto :**

- **Pionnier** : Premier ordinateur à intégrer un système d'exploitation entièrement basé sur une interface graphique.
- **Innovation clé** : Le passage du rappel (se souvenir de la commande) à la **reconnaissance** (reconnaître une icône ou un menu).
- **La métaphore du bureau** : Fichiers, dossiers, corbeille... des concepts du monde réel appliqués au numérique pour le rendre compréhensible.
- **Inventions associées** : La souris, l'impression laser, la mise en réseau Ethernet, la programmation orientée objet (Smalltalk). Un véritable "big bang" technologique.

**Xerox Alto en Image**

**Le grand-père de toutes nos interfaces.** Notez l'écran en mode portrait (pensé pour la bureautique, format A4), le traitement de texte graphique (WYSIWYG - What You See Is What You Get), et l'usage de la souris pour la sélection.



**Le Concept WIMP**

**L'acronyme qui a défini le paradigme de l'ère du desktop**

**:**

- **Windows (Fenêtres)** : Permet d'afficher et de gérer plusieurs applications ou contextes en même temps. La base du multi-tâches visuel.
- **Icons (Icônes)** : Représentations graphiques de fichiers, dossiers ou applications. Raccourcis visuels.
- **Menus (Menus)** : Listes d'actions et de commandes, rendant les fonctionnalités "découvrables" sans avoir à mémoriser des commandes.
- **Pointing device (Pointeur)** : La souris, qui permet une manipulation directe des éléments à l'écran (cliquer, glisser-déposer).

## 1984 : La Popularisation - Apple Macintosh

Steve Jobs visite le Xerox PARC, comprend le potentiel et décide de le rendre grand public.

- **La mission d'Apple** : Non pas inventer, mais **simplifier et intégrer** pour créer un produit accessible à tous, "l'ordinateur pour le reste d'entre nous".
- **Simplification radicale** : La souris passe de 3 boutons à 1 seul, rendant l'interaction évidente (on ne peut que cliquer).
- **Marketing de génie** : Le lancement avec la fameuse publicité "1984" réalisée par Ridley Scott lors du Super Bowl marque les esprits et positionne le Mac contre le "Big Brother" conformiste (IBM).

# **Macintosh System 1 (1984)**

**La première UI graphique à succès commercial.** On retrouve tous les éléments fondateurs : la barre de menu *en haut de l'écran* (une constante de macOS), des icônes au dessin soigné par Susan Kare, la corbeille, et des fenêtres qui se superposent.

[Macintosh 1984]

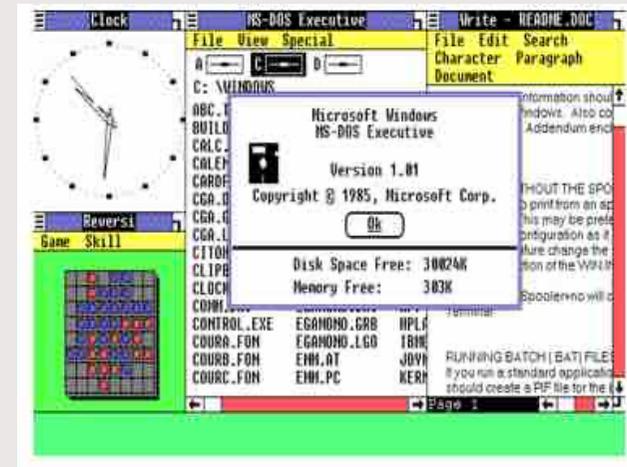


# 1985 : La Réponse - Microsoft Windows 1.0

Microsoft, qui développait des logiciels pour le Mac, lance sa propre interface graphique pour les PC compatibles IBM.

[Windows 1.0]

- **Approche différente** : C'était une surcouche graphique à MS-DOS, pas un OS intégré.
- **Interface moins aboutie** : Pour des raisons techniques et de brevets, les fenêtres ne pouvaient pas se superposer, elles se "tuilaient" (se plaçaient côte à côte). L'expérience était moins fluide.
- **Le début d'une longue compétition** : Malgré ses défauts, Windows 1.0 a posé la première pierre de la future domination de Microsoft sur le marché du desktop.



# Les Années 90 : Le Desktop Grand Public

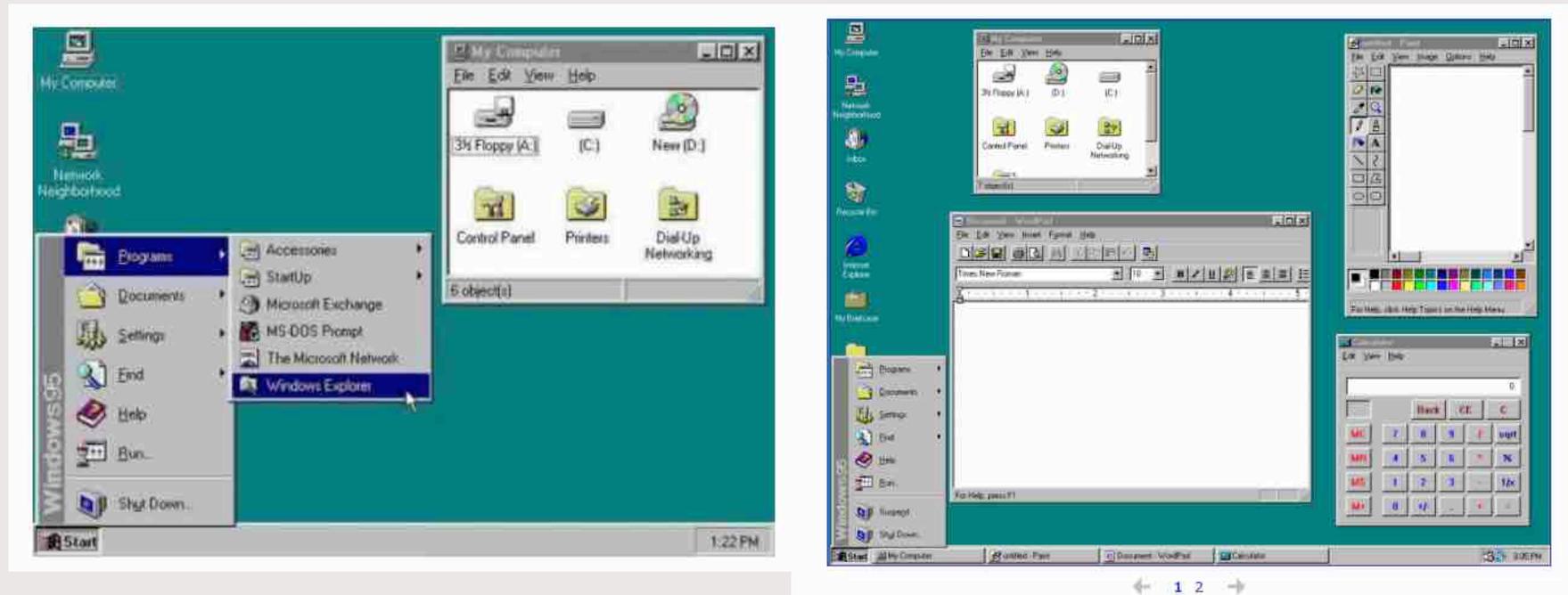
Windows 95 : Le point de bascule.

Windows 95 est l'OS qui a véritablement fait entrer l'informatique dans des millions de foyers. Il introduit des concepts qui sont encore la base de Windows aujourd'hui :

- **Menu Démarrer ( Start Menu )** : Un point d'entrée unique et centralisé pour lancer des applications et accéder aux fonctionnalités.
- **Barre des tâches ( Taskbar )** : Révolutionne la gestion des fenêtres en montrant en permanence les applications ouvertes et en permettant de passer de l'une à l'autre en un clic.
- **Standardisation massive** : Windows 95 est devenu le standard de fait, forçant les développeurs du monde entier à adopter ses conventions d'interface.

# Windows 95

L'interface qui a défini l'expérience PC pour une génération. L'icône "Poste de Travail", la "Corbeille", le bouton "Démarrer" sont devenus des éléments culturels.



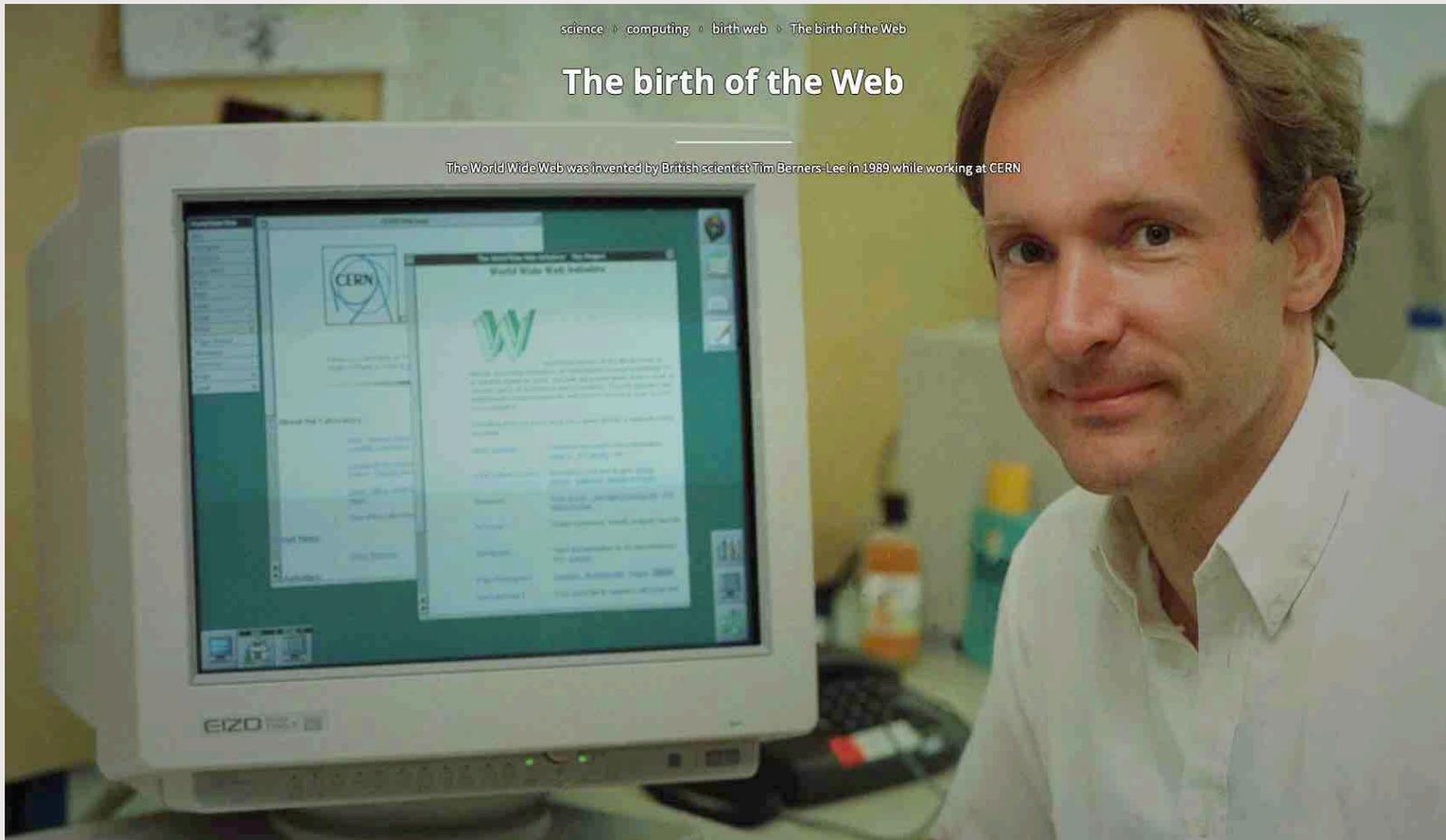
# **Les Années 90 : L'autre Révolution - Le Web**

1991 : Naissance du World Wide Web (Tim Berners-Lee, CERN)

- **Objectif initial** : Partager des documents de recherche entre scientifiques.
- **Interface initiale** : Textuelle, consultée via un navigateur en ligne de commande.
- **Primitive de navigation fondamentale** : **L'hyperlien**. C'est une rupture totale avec la navigation hiérarchique (dossiers/fichiers). On navigue de manière associative, de lien en lien.

## Le Premier Site Web (1991)

Un document sur sa propre nature. Il explique ce qu'est le World Wide Web, comment obtenir un navigateur, etc. L'interface est le document lui-même. C'est la quintessence de la fonction dominant la forme.



# L'Explosion Graphique du Web

Netscape Navigator (1994), puis Internet Explorer (1995) lancent la "guerre des navigateurs".

- **Support des images ( `<img>` tag )** : Le web devient visuel.
- **Les "hacks" de mise en page** : En l'absence de CSS, les développeurs détournent les **tables HTML** ( `<table>` ) pour créer des mises en page en colonnes. Un cauchemar pour la sémantique et l'accessibilité.
- **L'ère du "chaos créatif"** : GIFs animés clignotants, compteurs de visites, fonds d'écran en mosaïque, "sous construction". Le web est un Far West esthétique.
- **Skeuomorphisme** : Pour rendre le web moins intimidant, on utilise des boutons avec des effets de relief, imitant les objets du monde réel.

## Exemple Web des Années 90

**Le site de Space Jam (1996) est un fossile numérique parfaitement conservé.** Il est l'archétype du design de l'époque : fond étoilé, header graphique, navigation par images cliquables, et un chaos de couleurs et de polices.

[Space Jam website](#)

## **La Naissance des "Guidelines"**

Pour mettre de l'ordre dans le chaos applicatif, les éditeurs d'OS publient des guides de bonnes pratiques.

- **1987 : Apple Human Interface Guidelines**

Principes fondateurs : cohérence, manipulation directe, feedback, métaphores.

- **Années 90 : Microsoft Windows UX Guide**

L'objectif : faire en sorte qu'une application développée par un tiers ressemble et se comporte comme une application native de Windows.

**Le but est de créer un écosystème cohérent. L'utilisateur apprend à utiliser l'OS une seule fois, puis applique ces connaissances à toutes les applications.**

## Apple Human Interface Guidelines (HIG)

**Aujourd'hui, c'est une ressource en ligne complète qui dicte le design pour iOS, macOS, watchOS...**

Microsoft a son **Fluent Design System** et Google a **Material Design**. Ces guides sont la "bible" pour tout développeur visant une intégration parfaite sur une plateforme.

[Apple HIG](#)

# Le Tournant du Siècle : Le Mobile

Années 2000 : L'ère des smartphones "pré-iPhone"

- **Acteurs dominants** : BlackBerry (pour les pros, avec son clavier physique), Nokia avec Symbian OS (leader mondial), et Windows Mobile.
- **Paradigme d'interface** : Essentiellement un **desktop miniaturisé**. On retrouve un bouton "Démarrer", de petites icônes, des menus contextuels...
- **Interaction** : La navigation se fait principalement au **stylet** (écrans résistifs) ou avec un trackball/clavier. Ce n'est pas pensé pour le doigt.

## Exemple : Windows Mobile (2003)

Un bureau Windows dans sa poche. Remarquez la minuscule croix pour fermer une fenêtre, le menu "Démarrer" en haut à gauche... C'était puissant pour l'époque, mais terriblement peu pratique à utiliser.

### Windows Mobile

## 2007 : Le Big Bang - L'iPhone

Steve Jobs présente "3 produits Révolutionnaires en 1" :

1. Un iPod à écran large tactile.
2. Un téléphone mobile révolutionnaire.
3. Un terminal de communication internet.

**..Mais la vraie révolution était l'interface, qui unifiait ces trois concepts.**

# La Vraie révolution : le Multi-Touch

Les anciennes règles du WIMP volent en éclat.\*\*

**Fin de la manipulation indirecte** : Fini la souris.  
L'utilisateur **touche directement le contenu**. C'est un changement fondamental.

**Nouveaux verbes d'interaction** : Le dictionnaire de gestes Tap (clic), Swipe (faire défiler), Pinch-to-zoom (pincer pour zoomer), Long Press (appui long) devient le nouveau standard.

**Le Mobile-First est né** : On ne conçoit plus un logiciel pour ensuite le "réduire" sur mobile. On pense d'abord à l'expérience mobile, fondamentalement différente (contexte d'usage, taille d'écran, connectivité).

# L'iPhone OS 1 (2007)

Radicalement simple. Une grille d'icônes. Pas de système de fichiers visible. Pas de widgets. Pas de personnalisation. Chaque application est en plein écran. L'objectif est de réduire la charge cognitive à son strict minimum. Le matériel et le logiciel sont conçus en symbiose parfaite.



## La Réponse : Android & Material Design

Le défi de Google : Comment créer une expérience cohérente sur des milliers de téléphones de fabricants différents (Samsung, HTC, LG...) avec des tailles et résolutions d'écrans variées ?

- **2014 : Introduction de Material Design.** Ce n'est pas juste un "style", c'est un **système de design** complet.
- **Philosophie :** Le "papier et l'encre numériques". Les éléments d'interface ont des propriétés physiques. Ils sont sur des "feuilles de papier" numériques avec une élévation (une ombre), ils ont une masse et peuvent s'animer de manière cohérente.
- **Objectif :** Fournir aux développeurs un langage visuel et un ensemble de composants (widgets) unifiés pour créer des applications de qualité et cohérentes sur tout l'écosystème Android.

# Material Design de Google

Un langage visuel basé sur **des** métaphores physiques pour créer une hiérarchie et un sens. Une "Floating Action Button" (FAB) flotte au-dessus **des** autres contenus pour indiquer **l'**action principale.

Material Design

Introduction à Material Design

## **Conséquence : Le Responsive Design**

**Problème :** Un site web doit maintenant s'afficher correctement sur un écran 4 pouces, une tablette 10 pouces, un laptop 15 pouces et un moniteur 27 pouces.

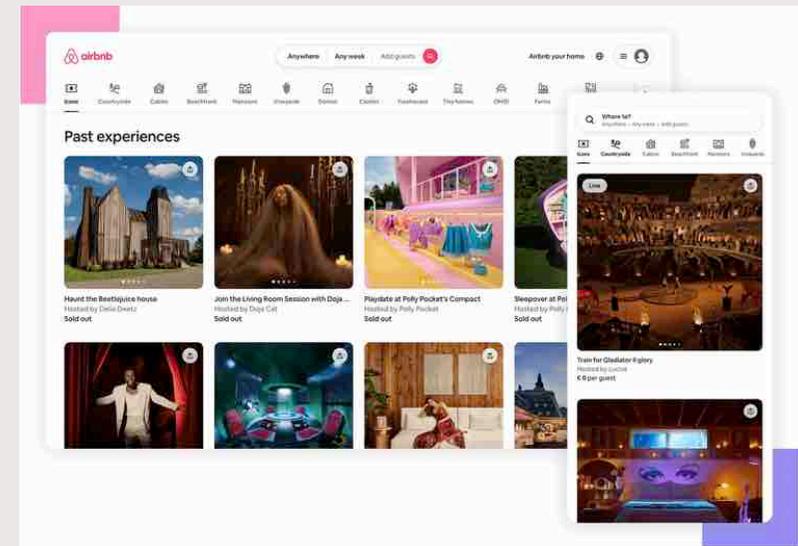
**Solution (Ethan Marcotte, 2010) : Le Responsive Web Design.** L'idée n'est pas de faire un site différent pour chaque taille, mais un seul site qui s'adapte.

- **CSS Media Queries** : Appliquer des styles différents en fonction de la largeur de l'écran.
- **Grilles fluides** : Utiliser des pourcentages ou des unités flexibles ( flexbox , grid ) plutôt que des pixels fixes.
- **Images flexibles** : Les images se redimensionnent pour tenir dans leur conteneur.

**Un seul code source, une multitude d'affichages.**

## Exemple de Responsive Design

**Le même site sur mobile, tablette et desktop.** Le contenu se réorganise : sur desktop, la navigation est en haut ; sur mobile, elle est cachée dans un menu "burger" ( $\equiv$ ) pour économiser de l'espace. Les colonnes se superposent.



# Synthèse de l'Évolution

Ère	Interface	Métaphore	Interaction	Exemple Paradigmatique
70s	CLI	Terminal	Texte, commandes	bash
80-90s	GUI	Bureau	Souris (WIMP)	Windows 95
90-00s	Web	Page/Document	Hyperlien, souris	Netscape
00s->	Mobile	Application	Touch & Gestures	iPhone OS
20s->	IA	Conversation	Langage naturel	ChatGPT

# Les Grandes Tendances Visuelles (1/3)

## 1. Skeuomorphisme (années 2000-2012)

Principe : Imiter l'apparence des objets du monde réel pour aider les utilisateurs à comprendre leur fonction.

Exemple : L'application "iBooks" d'Apple avec son étagère en bois, le "Notes" avec son papier jaune ligné et sa texture cuir.

But : Rassurer et familiariser l'utilisateur avec de nouveaux concepts numériques.



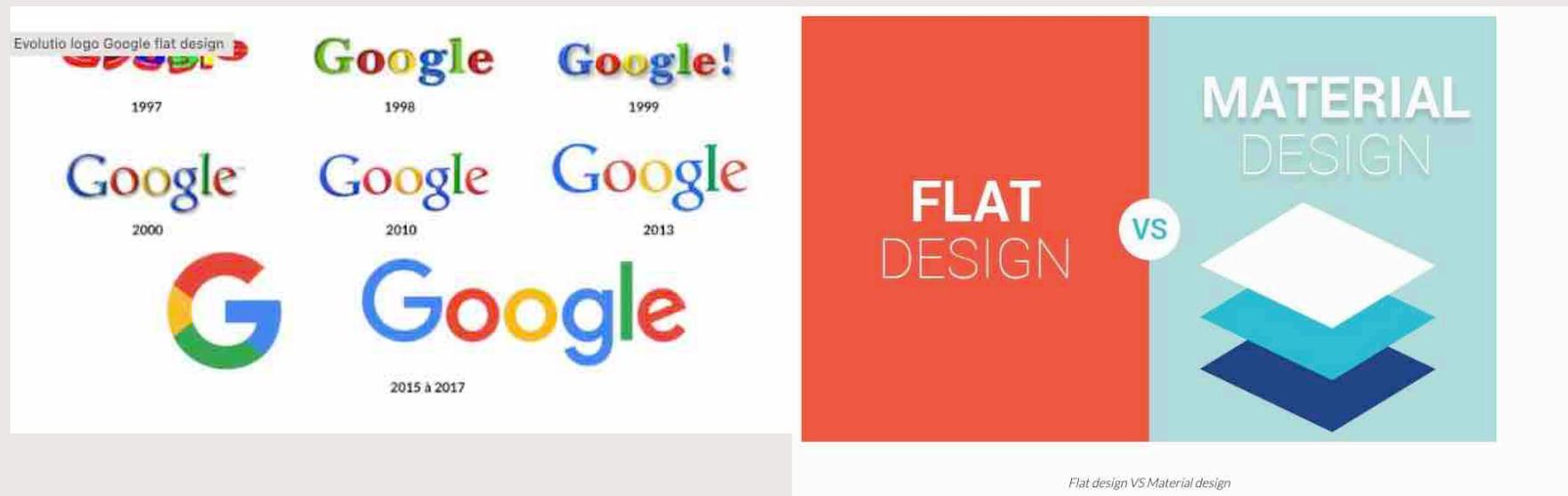
## Les Grandes Tendances Visuelles (2/3)

### 1. Flat Design (années 2012 ->)

*Principe* : Une réaction au skeuomorphisme. L'interface assume sa nature numérique. On se débarrasse des ombres, textures et gradients au profit de formes simples, de couleurs vives et de la typographie.

*Exemple* : Le passage d'iOS 6 à iOS 7, le design "Metro" de Windows 8.

**But** : Mettre le contenu au premier plan et créer une esthétique plus propre et moderne.



## Les Grandes Tendances Visuelles (3/3)

1. **Évolutions Récentes (Neumorphism, Glassmorphism)**  
*Principe* : Raffinements du Flat Design.
2. **Neumorphism (2019)** : Un "nouveau skeuomorphisme" soft. Les éléments semblent extrudés du fond, avec des ombres internes et externes douces. Tendance éphémère car difficile à rendre accessible.
3. **Glassmorphism (2020 ->)** : Utilise des arrière-plans flous et transparents, comme du verre dépoli. Crée une sensation de profondeur et de hiérarchie.  
*Exemple* : Le design de macOS Big Sur, les menus de Windows 11.



# **La Nouvelle Frontière : L'Ère de l'IA**

## **Un nouveau changement de paradigme est en cours.**

De la même manière que le tactile a remplacé la souris, le langage naturel est en train de devenir une nouvelle forme d'interaction primordiale.

- **De la manipulation directe...**

(Je clique sur "Fichier", puis "Enregistrer sous...", je nomme le fichier et je clique sur "OK")

- **...À l'interaction basée sur l'intention.**

(Je dis/écris : "Sauvegarde ce document dans mon dossier projet sous le nom de rapport final")

L'interface n'est plus un ensemble de commandes statiques, mais un **collaborateur** ou un **assistant** qui comprend l'objectif de l'utilisateur.

## Exemples d'Interfaces Augmentées par l'IA (IAI)

L'IA n'est plus juste en coulisses, elle devient l'interface elle-même.

### 1/ Interfaces Conversationnelles (CUI - Conversational UI)

- **Vocales :** Siri, Google Assistant, Alexa. Le hardware disparaît au profit de la voix.
- **Textuelles :** ChatGPT, `GitHub Copilot Chat`. L'interface principale est une zone de texte.

## 2/ Génération d'Interface à la volée

- **Code & UI :** Des outils comme `v0.dev` (Vercel) ou Galileo AI génèrent du code de composant UI (`React`, `Vue`) à partir d'une simple description textuelle.
- **Contenu :** Midjourney, DALL-E génèrent des images qui sont ensuite intégrées dans les interfaces.

### 3/ Interfaces Proactives et Personnalisées

- **\*\*Prédiction :\*** La "Smart Reply" de Gmail qui suggère des réponses, ou Spotify qui crée une playlist "Découvertes de la semaine". L'UI anticipe vos besoins.

## AI Interfaces examples

*De gauche à droite : un chatbot, un assistant vocal, et l'interface de v0.dev générant du code à partir d'un prompt.*

## Nouveaux Enjeux pour les Développeurs

### Le rôle du développeur d'interface évolue.

1. **Le "Prompt" comme nouvelle API :** Savoir décrire précisément une intention à une IA ( Prompt Engineering ) devient une compétence clé pour générer du code, des textes ou des images.
2. **Gestion de l'incertitude :** Une IA peut se tromper, "halluciner" des faits, ou renvoyer des résultats non-déterministes. Comment concevoir une UI qui gère ces cas ? (ex: "Je ne suis pas sûr de comprendre, voici 3 options...").
3. **Éthique et Transparence :** Comment indiquer clairement à l'utilisateur qu'il interagit avec une IA ? Comment expliquer les décisions ou recommandations de l'IA (la "boîte noire") ?
4. **Orchestration vs Construction :** Le développeur passe de "constructeur" de chaque pixel à "orchestrateur" de services IA qui construisent des parties de l'interface ?

**La question ouverte :** L'UI va-t-elle disparaître au profit d'une unique boîte de dialogue, ou va-t-elle devenir un "canevas intelligent" sur lequel l'IA et l'utilisateur collaborent ?