



Le « Vibe Coding » ou la programmation par l'intention

Guide pratique pour les enseignants

Ce guide rassemble des conseils, des concepts clés, du vocabulaire spécialisé et des exemples de rédactions de requêtes pour concevoir des applications web simples et interactives grâce à l'intelligence artificielle, sans avoir besoin de maîtriser la programmation.

Note : Si vous maîtrisez déjà la programmation, ce guide vous permettra d'aller encore plus loin et de mieux mesurer toute l'étendue des possibilités.

1 PLANIFICATION DU PROJET

Avant de rédiger la moindre consigne de code pour l'IA, prenez le temps de définir clairement votre besoin.

- **Séparer la planification du codage :** Il est fortement conseillé de procéder en deux étapes distinctes en ouvrant deux conversations séparées avec l'IA :
 - **Première conversation (Planification et conception) :** Définissez le concept, validez la faisabilité de l'idée et rédigez un cahier des charges.
 - **Deuxième conversation (Développement du code) :** Fournissez le cahier des charges finalisé pour générer la structure de base, puis testez et ajoutez les fonctionnalités une à une.
- **Évaluer le besoin réel :** Posez-vous la question : l'application est-elle vraiment nécessaire, ou un simple agent conversationnel (ex. un GPT personnalisé) suffirait-il?
- **Penser en termes d'intrants et d'extrants (Inputs / Outputs) :**
 - **Intrants :** Que va fournir l'élève ou l'enseignant à l'application (du texte, un fichier, des choix de réponses)?
 - **Extrants :** Que doit générer l'application (un résultat, une image, un fichier à télécharger, une animation)?
- **Définir le public cible :** Précisez le niveau scolaire et l'âge des élèves dans vos consignes pour adapter l'ergonomie, la taille des boutons et le vocabulaire de l'interface.
- **Procéder par étapes (découpage du projet) :** Évitez de tout demander en un seul coup pour un projet complexe :
 1. Rédigez le concept général ou un cahier des charges simple.
 2. Codez la structure de base.
 3. Testez l'application.
 4. Ajoutez ensuite les fonctionnalités secondaires une par une (ex. : d'abord le formulaire, puis le calcul, puis l'option d'exportation).
- **Valider la faisabilité :** Demandez directement à l'IA si votre idée est réalisable sous la forme d'une application web à page unique (Single Page Application).

Comprendre ce qu'une application web locale (exécutée directement dans le navigateur) peut ou ne peut pas faire permet d'éviter les pertes de temps.

Ce que ce type d'application web peut faire dans un cadre éducatif :

- **Interfaces interactives** : Écrans, boutons, formulaires, menus, tableaux de bord de classe.
- **Éléments visuels dynamiques** : Icônes, animations, émojis, jeux d'association ou de cartes.
- **Gestion des fichiers de l'utilisateur** : Importer un document, exporter en PDF ou en fichier texte.
- **Sauvegarde temporaire** : Conservation des données dans la mémoire du navigateur.
- **Multijoueur local** : Possibilité de créer des jeux au tour par tour ou en équipe sur un même écran (ex. : faire passer la tablette d'un élève à l'autre).

Ce qu'elle ne peut pas faire :

- Pas de base de données en ligne permanente.
- Pas de comptes d'utilisateurs ni de mots de passe sécurisés.
- Pas de fonctionnalités multijoueurs en temps réel sur des appareils différents connectés en réseau (sans serveur externe).

⚠ **Sécurité des données** : Le code d'une application locale est public et accessible dans le navigateur. Ne jamais y intégrer de données confidentielles ou personnelles sur les élèves (noms de famille, notes, identifiants).

Pourquoi et comment les données sont-elles conservées?

Les données (progression d'un jeu, pointage, réglages de l'enseignant) peuvent être enregistrées dans la mémoire temporaire du navigateur web (localStorage). Ces informations restent sauvegardées tant que la session du navigateur demeure active ou jusqu'à ce que le cache de l'appareil soit vidé. Comme l'application ne dispose pas d'un serveur distant pour sauvegarder les informations de façon permanente, la fermeture de l'onglet ou la réinitialisation du navigateur remet l'application à son état initial.

Gestion des médias (images et sons) :

- **Liens d'hébergement stables** : Vous pouvez ajouter une image à votre code à l'aide d'un lien d'hébergement stable. Un lien d'hébergement stable est une adresse web directe (URL) pointant vers un fichier multimédia déposé sur un serveur permanent et accessible en tout temps (ex. : document hébergé publiquement sur le Web, espace GitHub ou service d'illustration en ligne). Si le lien provient d'une source temporaire ou privée, l'image finira par afficher une erreur.
- **Images et illustrations vectorielles (SVG)** : Privilégiez le format vectoriel SVG pour les icônes, logos ou illustrations. Il s'intègre directement sous forme de texte dans le code source de l'application, éliminant ainsi tout risque de lien brisé.
- **Sons et musiques** : Intégrez des effets sonores légers pour donner une rétroaction immédiate, en utilisant des fichiers audio libres de droits hébergés sur des serveurs publics stables.

3

VOCABULAIRE ET CONSIGNES CLÉS À UTILISER AVEC L'IA

Pour obtenir une application fonctionnelle, voici le vocabulaire à privilégier dans vos requêtes pour préciser l'expérience utilisateur (UX).

Disposition et affichage

- **Pleine page (full page)** : L'application occupe tout l'écran sans marges inutiles.
- **Sans défilement (zero scrolling)** : L'ensemble des éléments reste visible sans avoir à faire défiler la page.
- **Conception adaptative (responsive)** : S'adapte aussi bien sur ordinateur que sur tablette ou téléphone.

Composants d'interface (UX) et exemples pédagogiques

- **Écran d'accueil (splash screen)** : Écran initial présentant les consignes et le titre. *Exemple : Afficher les règles d'un jeu de révision mathématique avec un bouton « Commencer la quête ».*
- **Conteneur (container)** : Zone qui regroupe visuellement un ensemble de contrôles ou de textes. *Exemple : Séparer visuellement les cartes de vocabulaire à glisser d'un côté et la zone de dépôt des définitions de l'autre.*
- **Fenêtre modale (modal)** : Fenêtre surgissante qui apparaît par-dessus l'interface pour demander une confirmation ou une saisie. *Exemple : Faire apparaître un message d'avertissement lorsque l'élève clique sur « Abandonner la partie » pour confirmer son choix.*
- **Infobulle (tooltip)** : Court texte explicatif qui apparaît au survol d'un élément. *Exemple : Afficher un indice grammatical ou la définition d'un mot difficile lorsque l'élève passe son curseur dessus.*
- **Notification éphémère (toast message)** : Petit message temporaire qui confirme une action. *Exemple : Afficher une alerte discrète « Réponse enregistrée! » ou « Bravo! +10 points » qui disparaît automatiquement au bout de deux secondes.*
- **Panneau de configuration enseignant (Teacher Dashboard)** : Zone réservée à la gestion de l'activité. *Exemple : Créer un panneau protégé par un mot de passe simple permettant à l'enseignant de choisir le nombre d'équipes, de régler la durée du chrono ou de copier-coller une liste de mots de vocabulaire personnalisée avant le cours.*

Contrôles et fonctions utiles

- **Génération d'URL personnalisée** : Demander à l'IA de créer un lien web contenant les paramètres réglés par l'enseignant pour que les élèves ouvrent directement la bonne version de l'exercice.
- **Avertissement avant fermeture** : Ajouter une alerte si l'élève tente d'actualiser ou de fermer la page pour éviter la perte accidentelle de travail.
- **Bouton d'impression / de réinitialisation** : Permet d'imprimer la réalisation de l'élève ou de remettre à zéro le compteur.

💡 Modèles de requêtes (Prompts types à copier-coller)

Prompt 1 : Validation de faisabilité (Étape 1)

« Je suis enseignant au [niveau scolaire]. Je souhaite créer une application web simple pour [objectif pédagogique, ex. réviser les fractions]. Les élèves devront [intrants, ex. glisser des éléments dans des cases] et l'application affichera [extrants, ex. un pointage et une animation de réussite]. Est-ce réalisable sous forme d'application web à page unique (Single Page Application) autonome, dans un seul fichier HTML sans serveur distant? »

Prompt 2 : Génération de la structure de base (Étape 2)

« Agis comme un développeur web expert en ergonomie éducative. Génère le code complet d'une application web à page unique (HTML, CSS Tailwind via CDN, JavaScript) tenue dans un seul fichier.

Spécifications :

- Interface adaptative, sans défilement (zero scrolling), lisible sur tablette.
- Inclure un écran d'accueil (splash screen) avec les règles du jeu et un bouton "Commencer".
- Utiliser des icônes SVG intégrées.
- Design sobre avec une palette de 2 couleurs principales. »

4 DESIGN ET EXPÉRIENCE UTILISATEUR (UX/UI)

Un bon design n'a pas besoin d'être complexe; il doit surtout être fonctionnel, sobre et accessible.

- **Sobriété visuelle** : Limitez votre palette de couleurs à 2 ou 3 teintes principales et choisissez une police de caractères très lisible. Évitez de surcharger l'écran d'éléments inutiles.
- **Rétroaction claire** : Chaque action de l'élève doit avoir une réponse visuelle ou sonore immédiate. Si un bouton est cliqué, un changement d'état doit être visible.
- **Messages d'erreur explicites** : Préférez une explication claire (ex. : « Veuillez entrer un nombre entre 1 et 10 ») à un message générique (« Erreur de saisie »).
- **Mécaniques d'engagement (max. 2 ou 3 par activité)** :
 - Barre de progression.
 - Système de points ou de niveaux.
 - Effets sonores ou visuels de célébration en cas de réussite.
 - ⚠ **Attention** : Veillez à ce que la rétroaction en cas d'erreur ne soit pas plus amusante ou gratifiante que la bonne réponse!
- **Accessibilité** :
 - Demandez à l'IA d'ajouter du texte alternatif descriptif (alt text) pour toutes les images afin d'assurer le soutien aux synthèses vocales.
 - Assurez-vous d'un contraste suffisant entre le texte et la couleur de fond.

5

ASTUCES POUR UN DÉVELOPPEMENT RÉUSSI ET SANS FRUSTRATION

- **Adopter une démarche itérative** : Obtenir une application parfaite dès la première tentative est très rare. Préparez-vous à tester, corriger et ajuster progressivement.
 - **Guider l'IA avec des visuels** : Vous pouvez fournir une capture d'écran annotée ou un croquis fait à la main pour expliquer exactement où placer les boutons ou quel style imiter.
 - **Expliquer le problème, pas seulement le résultat** : Si un élément ne fonctionne pas, décrivez à l'IA le comportement observé par rapport au comportement attendu.
 - **Réinitialiser la conversation si nécessaire** : Si l'IA commence à tourner en rond ou à faire des erreurs répétées, ouvrez une nouvelle discussion, donnez-lui le code actuel et réexpliquez votre besoin de manière sobre.
-  **Attention au piège du coût irrécupérable (sunk cost fallacy)** : Si une fonctionnalité bloque votre projet depuis des heures, évaluez s'il vaut mieux la simplifier ou l'abandonner au profit d'une version plus sobre mais fonctionnelle.
- **Attribution et transparence** : Pensez à inclure dans le pied de page de votre application une note d'attribution mentionnant votre nom ainsi que l'utilisation de l'intelligence artificielle pour la conception.