Vibe coder avec CAPTE

CAPTE est une méthode de vibe coding créée par **Clément Cahagne**. Cinq lettres, cinq questions, posées dans l'ordre. De quoi transformer une demande vague en un prompt qui produit ce que vous aviez en tête, et savoir quoi corriger quand ce n'est pas le cas.



Clément Cahagne — pauuse
Ingénierie pédagogique & IA appliquée à la formation
in linkedin.com/in/clementcahagne ✉ clement.cahagne@pauuse.fr

Le principe

Tout ce que vous ne décidez pas, l'IA le décide à votre place — la couleur du fond, la police, la durée d'une animation — et elle ne vous préviendra pas. Prompter, ce n'est donc pas trouver la formule magique. C'est ne rien laisser d'important au hasard. CAPTE est la liste de ce qui est important : cinq lettres, cinq questions, dans un ordre qui n'est pas décoratif.

> Sur une petite application, un prompt CAPTE complet tient en quinze lignes, prend cinq minutes de réflexion et vous évite des heures de correction !

Deux repères pour ne jamais hésiter

1. L'ordre va **du pourquoi vers le comment**.
2. **A décrit ce qui ne bouge pas, P décrit ce qui change.**

CAPTE en un coup d'œil

LETTRE	LA QUESTION	CE QUE VOUS Y METTEZ
C — Contexte	À quoi ça sert, pour qui, dans quelle situation ?	L'objectif, l'utilisateur, le moment d'usage
A — Apparence	À quoi ça ressemble quand rien ne bouge ?	Ambiance, couleurs, typographie, formes
P — Parcours	Que fait et voit l'utilisateur, écran par écran ?	Le scénario **et** les croquis
T — Technique	Y a-t-il une contrainte ?	Rien, sauf contrainte réelle
E — Expertise	Que savez-vous que l'IA ne peut pas deviner ?	Vos contenus, règles, messages, un exemple

Les cinq lettres

C — Contexte *À quoi ça sert, pour qui, dans quelle situation ?*

- **À écrire** : l'objectif en une phrase, la personne qui utilise, le moment et l'appareil.
- **Pourquoi en premier** : tout le reste en découle. Le même « lanceur de dés » ne donne pas la même application pour une soirée entre amis et pour un jeu pédagogique en classe.
- **L'erreur classique** : « Crée-moi une application de… » décrit l'objet, pas son usage. Dites-lui à quoi elle sert.

A — Apparence *À quoi ça ressemble quand rien ne bouge ?*

- **À écrire** : l'ambiance en trois mots, les couleurs, la typographie, la forme des éléments
- **L'erreur classique** : « moderne et épuré » ne veut rien dire pour une IA : c'est la moyenne de ce qu'elle a vu, du gris et du bleu. Donnez des codes HEX, citez une référence, ou nommez un univers.

P — Parcours *Que fait et voit l'utilisateur, écran par écran ? Deux couches, détaillées page 3.*

- **Le scénario s'écrit** : les actions, les enchaînements, le temps, les conditions, les mots exacts.
- **Le croquis se dessine** : la disposition, les tailles, les zones, l'enchaînement des écrans. Une feuille et un stylo suffisent. Ou un outil comme Excalidraw ou Claude Design pour créer les wireframes.
- **En un mot** : le croquis dit **où**, le scénario dit **quand**.

T — Technique *Y a-t-il une contrainte technique ? Une stack à privilégier ?*

C'est la seule lettre où « je ne sais pas » est une bonne réponse : à condition de le dire au lieu de laisser le vide. Vous ne décrivez pas une technologie mais une **contrainte**, en français (« ouvrable d'un double-clic », « utilisable sur téléphone ») : le choix technique est le métier de l'IA. Par défaut, vous déléguez :

```
Je n'ai pas de contrainte technique. Choisis le langage ou le framework
le plus adapté et le plus simple qui fonctionne pour cette demande,
et précise-moi en une phrase ce que tu as choisi.
```

E — Expertise (la vôtre) *Qu'est-ce que je sais que l'IA ne peut pas deviner ?*

Ce que l'IA ne sait pas mais devrait savoir pour faire bien. Quatre tiroirs :

1. **Vos contenus** — ce qui doit apparaître tel quel : textes, tarifs, questions, noms.
2. **Vos règles** — tout ce qui décide : calculs, seuils, conditions, cas particuliers.
3. **Vos messages** — ce que l'application dit à l'utilisateur, dans vos mots.
4. **Vos exemples** — un cas concret traité de bout en bout, avec le résultat attendu. Un exemple chiffré vaut dix règles.

Le parcours, écran par écran

Le P en deux couches

Couche 1 — le scénario (ce que l'application fait). Écran par écran : son nom et son rôle en une phrase, les actions possibles, le déclencheur du passage à l'écran suivant, les **états** (chargement, attente, succès, erreur, écran vide), les durées et animations, les règles de calcul et les messages exacts.

Presque personne n'écrit les états. Ce sont eux qui provoquent presque tous les bugs.

> ✗ « L'utilisateur consulte ses résultats. » — peut rester vague indéfiniment.

> ✓ « Écran unique. Au chargement, le dé affiche 1, immobile. Au clic sur LANCER, il roule 1 seconde et le bouton devient inactif. À la fin du roulement, le résultat s'affiche, le dé grossit de 30 % pendant une demi-seconde. »

Couche 2 — le croquis (où se trouvent les choses). Un croquis par écran, joint au prompt en image : position des éléments, hiérarchie des tailles, découpage en zones nommées, une flèche par transition avec l'action écrite dessus. Ce que vous **ne** dessinez **pas** : les couleurs (elles appartiennent au A) ; les durées, états, conditions et messages (ils appartiennent au scénario). 7 règles à respecter :

1. **Un cadre = un écran**, numéroté.
2. **Nommez chaque zone en toutes lettres** : « champ de texte », « bouton LANCER »
3. **Respectez les proportions** : l'IA lit la hiérarchie visuelle avant les mots.
4. **Une flèche entre deux écrans**, avec l'action écrite dessus.
5. **Les états ne se dessinent pas, ils s'écrivent.** Un dessin ne sait pas dire « pendant une seconde »
6. **Noir et blanc obligatoire.** La couleur appartient au A.
7. **Écrivez lisiblement.** L'IA lit votre écriture, au sens propre.

Les trois lignes à joindre au prompt

Un croquis envoyé sans mode d'emploi produit parfois une application… en noir et blanc.

```
Les images jointes sont les croquis des écrans 1 à N.
Respecte la disposition des éléments et leur hiérarchie de taille.
En cas de contradiction entre un croquis et le texte, le texte fait foi.
Ne reproduis pas le style du croquis : il est volontairement en noir et blanc,
l'apparence est décrite dans la section APPARENCE.
```

Lire le résultat et corriger

Nommer l'écart, pas le ressenti. On place le croquis à côté de l'écran, on compare, puis on nomme.

RESSENTI (INEXPLOITABLE)	ÉCART NOMMÉ (EXPLOITABLE)
« Ce n'est pas beau »	« Le bouton est en haut alors que le croquis le place sous le dé »
« Ça bug »	« Au clic, rien ne se passe : le dé reste sur la même face »
« C'est trop rapide »	« Le roulement dure moins d'une seconde, alors qu'il en faut une pleine »

Les trois gestes de correction :

1. **Une seule chose à la fois.** Trois demandes dans un même message, et l'IA réécrit tout — y compris ce qui marchait.
2. **La phrase qui protège le reste** : « Ne modifie que le bouton. Laisse tout le reste strictement identique. »
3. **Recadrer quand ça part en vrille** : renvoyez le croquis, redonnez la ligne TECHNIQUE et la phrase de CONTEXTE. L'IA a perdu le fil, pas la compétence.

Les 5 idées à retenir

1. Il n'y a jamais de case vide : tout ce que vous ne décidez pas, l'IA le décide.
2. CAPTE va du pourquoi vers le comment — A décrit ce qui ne bouge pas, P ce qui change.
3. Le parcours se décrit sur deux couches : l'image dit **où**, le texte dit **quand**.
4. Le T reste vide par défaut ; on ne le remplit que s'il y a une vraie contrainte.
5. Le E, c'est votre expertise, pas celle de l'IA — et il apparaît dès qu'il y a une deuxième chose.