Eigenflow

GUIDE TESTÉ · CLAUDE CODE

Faire des reels animés avec Claude Code

Les 6 étapes refaites pour de vrai le 2 novembre 2026 : les prompts exacts, les durées, la v1 et la v2, et le piège du dossier.

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté MONTEUR. Voici le guide du reel : les six étapes dans l'ordre, les trois prompts exacts qu'on a tapés, ce que Claude a fait à chaque fois et en combien de temps. Puis un piège sur l'emplacement du dossier, une skill à écrire pour aller plus loin que notre v2, et les limites du test.

On a tout refait pour de vrai le 2 novembre 2026, entre 13 h 33 et 13 h 52, sur un Mac, dans un dossier vide. Outil : Claude Code avec Claude Opus 5.5, sur l'abonnement Claude, sans clé d'API. Sujet du reel d'exemple : pourquoi la nourriture d'avion a si peu de goût.

Les phrases de Claude citées ici sont recopiées mot pour mot de nos traces. Ce qu'on n'a pas pu vérifier est écrit.

À savoir avant de lire

• Claude a travaillé 15 min 18 s en tout, pour trois prompts et deux vidéos.

• L'installation a pris 5 min 35. Rien n'a été installé en global : Remotion et edge-tts sont allés dans le dossier.

• La v1 est sortie en 2 min 22. Elle dure 43 s et ressemble à une suite de 7 diapos.

• La v2 est sortie en 7 min 21, après une seule remarque. Elle dure 37 s et compte 47 temps forts, chacun calé sur un mot de la voix.

• À la fin, le dossier pèse 955 Mo pour 20 375 fichiers. Le projet lui-même tient en 19 fichiers, le reste ce sont les outils.

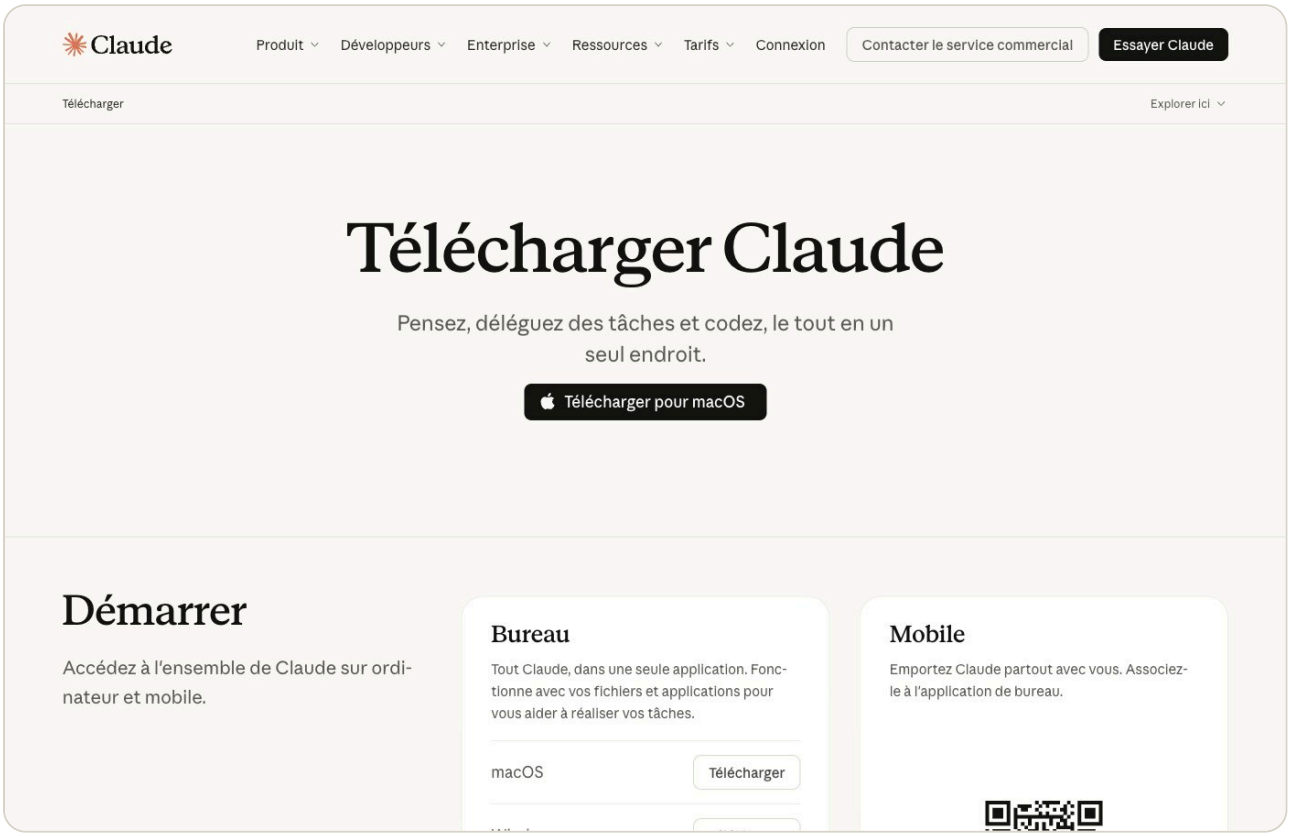
• Pendant le test, le rendu bloquait dans notre dossier rangé sous le Bureau, et Claude en a donné une mauvaise cause. On te conseille Documents (partie 7).

• Au prix de l'API, ces trois tours auraient coûté 2,83 dollars. Sur l'abonnement, rien à payer en plus : ça se déduit du quota.

1. Télécharge l'appli desktop

Le reel dit d'éviter le navigateur. La raison est dans la documentation de Claude Code. Dans le navigateur, Claude Code tourne dans le cloud d'Anthropic et travaille sur des dépôts de code. Dans l'appli, une session Local tourne sur ton ordinateur, avec un accès direct à tes fichiers. Pour monter des vidéos avec des outils installés chez toi, il te faut la seconde.

La page officielle : claude.com/download. L'adresse claude.ai/download y renvoie. Relevée le 2 novembre 2026 à 14 h 22, la page française affiche un bouton « Télécharger pour macOS ». La documentation de Claude Code donne aussi les installeurs Windows (x64 et ARM64) et une version bêta pour Linux (Ubuntu et Debian).



La page officielle de téléchargement, claude.com/download, en français. Capture du 2 novembre 2026 à 14 h 22, à 1400 × 900.

Une fois l'appli installée : lance Claude, connecte-toi, clique sur l'onglet Code. D'après la doc, l'appli contient déjà Claude Code, donc pas besoin d'installer la version terminal à côté. Pour l'onglet Code, il faut un abonnement payant.

Chez nous, l'appli était en version 4.4212.0. On ne pouvait pas la piloter depuis notre session de test, alors on a joué la même session en ligne de commande. Le détail est dans la partie 9.

2. Crée le dossier « Monteur vidéo IA »

Dans le Finder, clic droit, « Nouveau dossier », et appelle-le Monteur vidéo IA. Range-le dans Documents (la partie 7 explique pourquoi).

Ce dossier contient tout le projet. Claude y écrit le script, la voix, l'animation et la vidéo finie, et c'est là que tu rangeras ta skill (partie 8). Voilà ce qu'il y avait dedans à la fin de notre test :

- `src/script.json` : le texte de chaque scène et ses temps forts ;
- `src/Reel.tsx` : l'animation, 247 lignes de React ;
- `scripts/voix.py` : la voix off et l'horodatage des mots, 100 lignes ;
- `public/voix/` : 7 fichiers MP3, un par scène ;
- `out/reel-nourriture-avion.mp4` : la vidéo ;
- `node_modules/` (19 972 fichiers) et `.venv/` (384 fichiers) : les outils installés par Claude.

Dans le test, le dossier s'appelait « Monteur video IA », sans accent. On l'a créé vide à 13 h 33.



Illustration tirée du reel, pas une capture : le dossier créé dans Documents, puis son contenu. Dans le vrai test, le fichier de voix final s'appelle voix.py, et la skill se range dans un sous-dossier .claude/skills (partie 8).

3. Ouvre une session Local sur ce dossier

Dans l'onglet Code, commence une nouvelle session. Sous la zone de saisie, il y a une rangée de réglages. Le premier choisit l'environnement : prends Local, qui tourne sur ta machine avec un accès direct à tes fichiers. Celui d'à côté, « Sélectionner un dossier », choisit où Claude travaille : prends Monteur vidéo IA.

Tape ensuite ton premier message. À partir de là, Claude lit et écrit dans ce dossier, et lance des commandes sur ton Mac.

Un détail vu dans nos traces : dès sa deuxième commande, Claude a listé les deux dossiers au-dessus du sien et tenté d'y lire des fichiers .md. Il n'y en avait aucun. Garde quand même ce dossier loin de tes papiers privés.

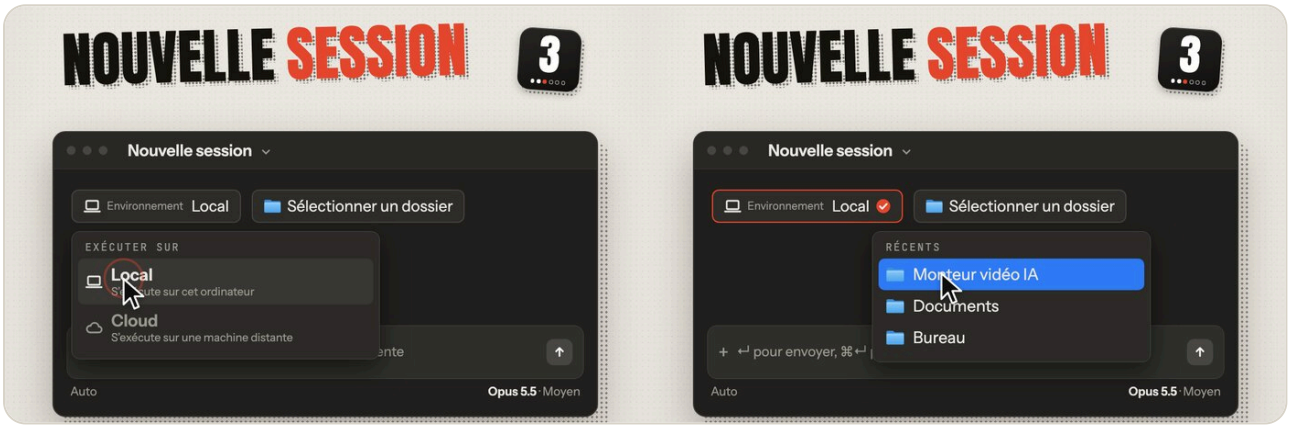


Illustration tirée du reel, pas une capture : l'environnement Local, puis le dossier. Les libellés « Environnement », « Local » et « Sélectionner un dossier » figurent bien dans les textes français de l'appli, version 4.4212.0.

4. Demande-lui de vérifier ton installation

Pour animer, faire la voix et rendre une vidéo, Claude a besoin d'outils. Laisse-le faire l'inventaire. Le prompt exact du test :

> **À COLLER**
>
> ```
> Vérifie mon installation : ce qu'il te faut pour animer, faire la voix off et rendre une vidéo verticale. Installe ce qui manque et guide-moi. Ne change rien en dehors de ce dossier : si un outil manque en global, dis-le-moi au lieu de l'installer.
> ```

Garde la dernière phrase. Avec elle, Claude n'a jamais tenté d'installer quoi que ce soit pour tout l'ordinateur : notre garde-fou, réglé pour refuser ce genre de commande, ne s'est pas déclenché une seule fois.

Durée : 5 min 35, de 13 h 36 à 13 h 41, avec 11 commandes.

Déjà installé sur notre Mac, il a trouvé :

• Node 22 (v22.22.2) et npm, pour faire tourner l'animation ;

• ffmpeg 8 (8.0.1) et ffprobe, pour encoder et mesurer la vidéo ;

• Python 3.14 (3.14.5) et uv 0.9.26, pour la voix ;

• ImageMagick 7.1.2, pour assembler des images.

Dans le dossier, il a installé :

• Remotion 4.0.540, une bibliothèque qui fabrique des vidéos en React, avec React 19.3.1, TypeScript 7.0.2 et son propre Chrome sans fenêtre, dans `node_modules/` ;

• edge-tts 7.3.1, pour la voix off, dans un environnement Python à part (`.venv/`) créé avec uv.

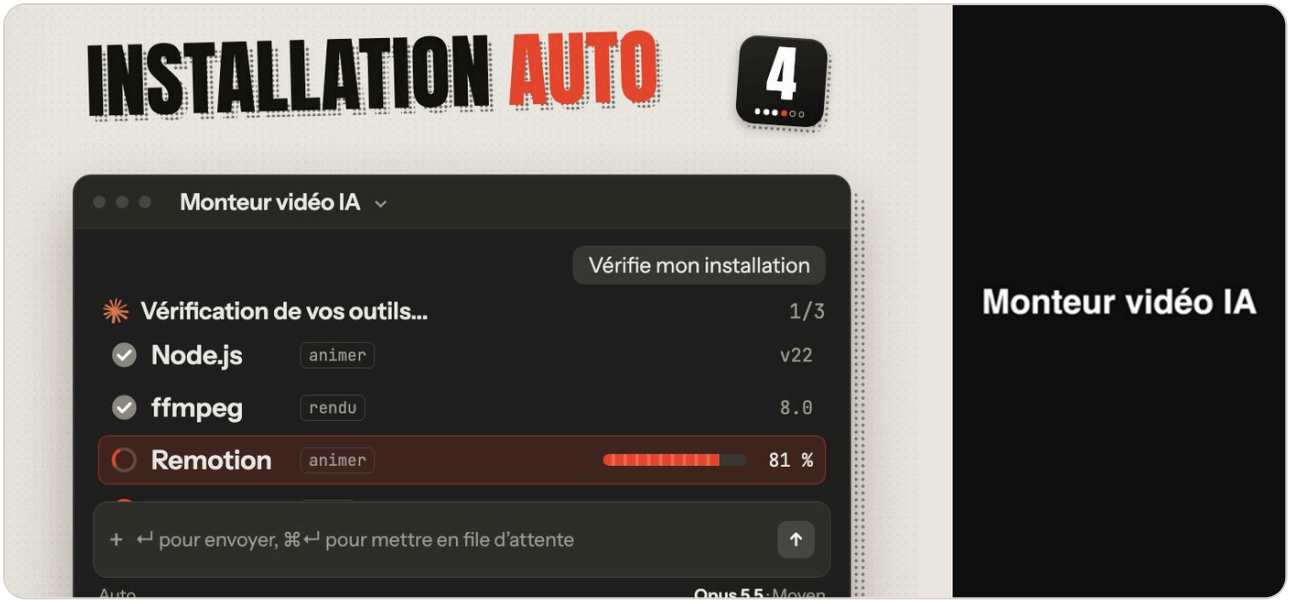
Ces deux outils, il les a choisis seul. Puis il a tout essayé de bout en bout : une phrase lue par la voix fr-FR-HenriNeural, une composition verticale 1080 × 1920 avec un titre animé, un rendu MP4. Le rendu a bloqué dans le dossier (partie 7). Sur une copie, il est passé : 5 s, image H.264 et son AAC. Claude a ensuite extrait une image de la vidéo et l'a regardée pour vérifier que le texte s'affichait.

Il a proposé deux ajouts sans les installer. Whisper, pour caler les sous-titres, « environ 1 à 2 Go avec le modèle ». ElevenLabs, pour une voix plus naturelle, mais « aucune clé n'est configurée ». Il a aussi prévenu : « Licence Remotion : gratuite pour un particulier ou une petite structure, payante au-delà de 3 personnes. » On a relu la licence de Remotion le 2 novembre : gratuite pour un particulier et pour une entreprise de 3 salariés au plus. Il a vu juste.

Sa conclusion sur notre Mac : « Rien d'indispensable ne manque en global. »

On a comparé ce qui est installé en global sur le Mac (Homebrew, npm, pip) avant et après le test : identique. Hors du dossier, seuls des caches ont bougé (celui de uv, celui de npm), plus une copie de travail de 957 Mo dans le dossier temporaire de Claude.

Sur un Mac où Node ou ffmpeg manquent, ce prompt demande à Claude de te le dire au lieu de les installer. On ne l'a pas vu faire : chez nous, tout était déjà là.



À gauche, illustration tirée du reel. À droite, la vraie image que Claude a extraite de son rendu de test de 5 s, puis lue pour vérifier son travail.

5. Donne le sujet en une ligne

Le prompt exact du test :

> **À COLLER**
>
> `Fais un reel sur pourquoi la nourriture d'avion a si peu de goût.`

Dans le reel, la voix dit « est si fade ». Le test, lui, a été lancé avec la phrase ci-dessus.

Durée : 2 min 22, de 13 h 42 à 13 h 44.

Ce qu'il a produit, dans l'ordre :

1. Un script de 7 scènes, écrit directement dans `src/script.json` : l'accroche « Zéro goût ? », une cabine pressurisée comme à 2 000 m, moins de 20 % d'humidité, –30 % sur le salé et le sucré, 85 dB de moteurs, l'umami, plus de sel. Un emoji et un gros chiffre par scène.

2. La voix, avec un fichier `scripts/voix.sh` qui enregistre une piste par scène : edge-tts, voix fr-FR-HenriNeural, débit +8 %. Environ 40 s de voix.

3. L'animation Remotion : un ciel en dégradé avec des nuages qui défilent, une barre de progression, l'emoji qui arrive avec un effet de ressort, un sous-titre qui s'allume mot par mot.

4. Le rendu, en MP4 1080 × 1920.

5. Un contrôle. Il a extrait 5 images (à 2, 9, 18, 30 et 41 s), les a montées en planche et les a regardées, puis une de plus à 32 s. Il a mesuré le son, –25,7 dB de moyenne, l'a jugé trop bas pour Instagram et l'a normalisé à –14 LUFS. Personne ne le lui avait demandé.

Résultat : une v1 de 43 s, avec voix et sous-titres. Claude dit lui-même où elle pèche : « Les sous-titres s'allument mot par mot, mais leur rythme est estimé d'après la longueur des mots, pas calé exactement sur la voix. »

On a extrait la v1 à une image par seconde pour la regarder en entier. Chaque scène reste figée 5 à 7 s, et la voix laisse des blancs de 0,9 à 1,3 s entre les scènes. On scrolle avant la deuxième.



La v1 à une image par seconde, de 0 à 19 s. « Zéro goût ? » reste 6 s à l'écran, « 2 000 m » 5 s. Seuls les mots de la phrase en gris s'allument.

6. Fais-lui une remarque de monteur

Regarde ta v1 comme un monteur et dis ce qui cloche, avec des secondes et une règle chiffrée. Notre remarque, sur le rythme seulement :

> **À COLLER**
>
> De 0 à 6 s, on voit la même image : l'avion, « Zéro goût ? » et la phrase entière en gris dont les mots s'allument un par un. Toutes les scènes font pareil pendant 5 à 7 s. Ça fait une suite de diapos et on scrolle avant la deuxième. Il faut qu'un nouvel élément arrive à l'écran au moins toutes les 1,5 seconde.

Garde la dernière phrase telle quelle. Le début, remplace-le par ce que tu vois dans ta propre v1.

Durée : 7 min 21, de 13 h 45 à 13 h 52, avec 20 commandes.

Ce qu'il a changé :

1. Les mots horodatés. Claude a trouvé dans edge-tts une option, WordBoundary, qui renvoie l'heure exacte de chaque mot prononcé. Les sous-titres sont donc calés sur la voix, sans Whisper.

2. 47 temps forts. Il a réécrit le script : chaque scène devient une suite d'éléments, chacun accroché à un mot (par exemple « d'avion ») et typé. Il y a dix types : icône, gros texte, compteur, jauge, barres, paire, rangée, étiquette, tampon, icône qui traverse l'écran.

3. Un contrôle automatique. `scripts/voix.py` remplace `voix.sh`. Il fait la voix, place les temps forts, coupe le silence en fin de phrase, et affiche « TROP LONG » dès qu'un passage dépasse 1,5 s sans nouvel élément. Au premier essai, il en a signalé 9, jusqu'à 2,38 s. Une fois les silences de fin de scène coupés, le pire tombait à 1,49 s. Claude a ajouté un temps fort pour garder de la marge : 1,41 s.

4. L'animation refaite : une entrée différente par type, un compteur d'altitude qui monte jusqu'à 2 000 m, une jauge d'humidité qui descend à 18 %, des barres salé et sucré qui rétrécissent, un compteur de décibels, des sous-titres par groupes de 3 mots au plus avec le mot prononcé en jaune, et un léger zoom à chaque nouvel élément.

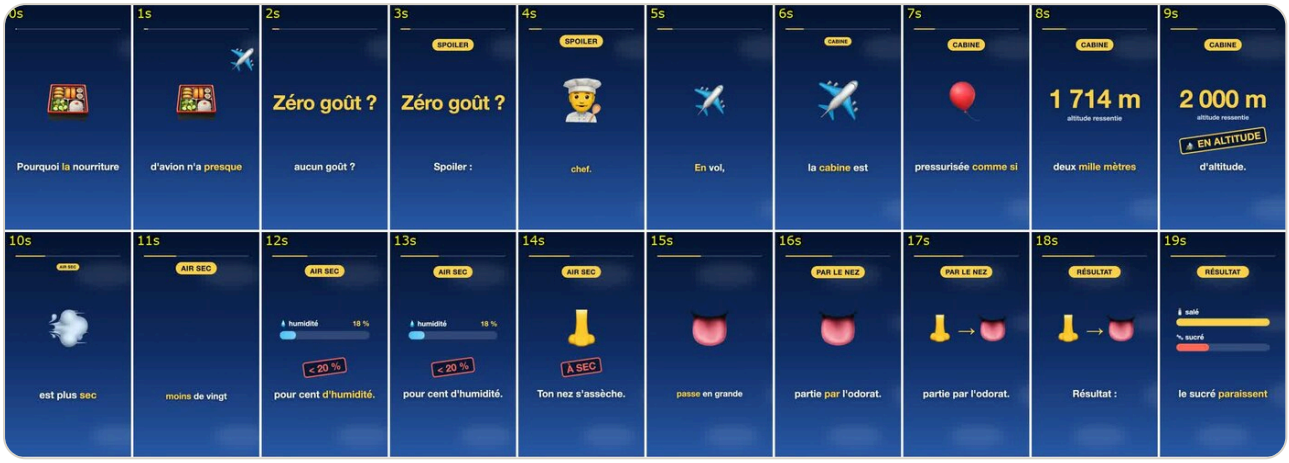
5. Le contrôle image par image. Il s'est monté une planche des 6 premières secondes (une image toutes les 0,5 s), une planche de toute la vidéo (une image par seconde), puis trois planches sur les changements de scène. Il les a regardées et a corrigé ce qu'il y voyait.

Ses mots sur ces corrections : « La montagne remplaçait le compteur avant qu'il n'atteigne 2 000 m. Des tampons longs débordaient de l'écran. L'espace manquait devant « ? » et dans « 1 182 m », car la police n'avait pas le bon caractère. »

Résultat : une v2 de 37 s, 6 s de moins que la v1 grâce aux silences coupés, avec le son toujours à −14 LUFS. Claude résume : « Il y a 47 temps forts au lieu de 7 diapos. » Puis : « L'élément le

plus long reste 1,41 s à l'écran avant le suivant. Une vérification automatique le contrôle à chaque génération. »

Sur notre planche de la v2, l'image change d'une seconde à l'autre. Deux images de transition sont presque vides : à 11 s, il ne reste que l'étiquette « AIR SEC », et à 32 s un élément encore minuscule.



La v2 à une image par seconde, de 0 à 19 s : plateau-repas, avion, « Zéro goût ? », étiquette « SPOILER », chef, compteur d'altitude. L'image presque vide à 11 s est une transition.



La planche que Claude s'est montée pendant l'étape 6 : les 6 premières secondes, une image toutes les 0,5 s, avec l'heure sous chaque image. Il l'a lue avant de continuer.



Un des trois défauts repérés par Claude sur ses planches. À gauche, à 8,9 s, la montagne a déjà pris la place du compteur d'altitude. À droite, après correction, à 8,5 s, 8,9 s et 9,3 s : le compteur atteint 2 000 m et reste à l'écran.

7. Le piège du dossier

Dès l'étape 4, Remotion a refusé de rendre la vidéo dans notre dossier. Il restait bloqué au chargement avec ce message : « A delayRender() … was called but not cleared after 28000ms ».

Claude a cherché. Il a copié le projet dans son dossier temporaire, sous un nom sans espace, et là le rendu passait. Sa conclusion : « Remotion, l'outil d'animation, ne charge pas le projet parce que le nom « Monteur video IA » contient des espaces. » Il a proposé de renommer le dossier en Monteur-video-IA et demandé deux fois notre feu vert. Le test ne prévoyait pas de tour pour répondre. Tous les rendus du test sont donc passés par une copie de 957 Mo, et Claude n'a plus retenté de rendu sur place.

Son diagnostic était faux. On l'a vérifié le jour même, pendant le test :

• le même projet, copié ailleurs dans un dossier nommé « avec espace », rend sans problème ;

• une copie au chemin exact du test, espaces compris, recréée hors du Bureau (sous /private/tmp), rend aussi.

On en a d'abord conclu que l'emplacement sous le Bureau bloquait, peut-être à cause de la protection que macOS met sur ce dossier. Puis, vers 14 h 35, on a refait l'essai. Une copie du projet dans Documents, avec des espaces dans le nom, charge. Et le dossier d'origine, au même endroit sous le Bureau, charge aussi, y compris depuis une session Claude Code lancée avec les réglages du test. Depuis notre terminal, il rend même 2 s de vidéo sur place. Le blocage vu à l'étape 4 ne s'est pas reproduit une heure plus tard.

On ne connaît pas la cause. L'espace dans le nom est exclu, et l'emplacement seul ne suffit pas à l'expliquer.

Ce qu'on en tire :

• crée le dossier dans Documents : on l'y a vu charger, et le Bureau est le seul endroit où on l'a vu bloquer ;

• si le rendu bloque avec ce message, ne renomme pas ton dossier sur la seule parole de Claude ;

• demande-lui plutôt de tester une copie du projet ailleurs, avec le même nom, puis relance un rendu sur place un peu plus tard.

8. Pour aller plus loin que la v2 : écris ta skill

Pour Claude Code, une skill est un fichier SKILL.md de consignes qu'il relit quand la tâche s'y prête. D'après la doc, une skill de projet se range dans `.claude/skills/nom-de-ta-skill/SKILL.md`, à l'intérieur du dossier. Sa description dit à Claude quand s'en servir, et tu peux aussi l'appeler à la main en tapant /monteur-reels (le nom de son dossier).

Ce prompt fait écrire à Claude les règles qui ont marché dans notre test :

À COLLER

```
Écris dans ce dossier une skill, .claude/skills/monteur-reels/SKILL.md, avec une description qui dit de s'en servir pour chaque reel, et ces règles : un nouvel élément à l'écran au moins toutes les 1,5 s, contrôlé par un script qui signale TROP LONG ; chaque apparition calée sur l'heure du mot prononcé (WordBoundary d'edge-tts) ; sous-titres par groupes de 3 mots au plus, mot prononcé surligné ; silence de fin de phrase coupé ; son normalisé à −14 LUFS ; une planche d'une image par seconde extraite et relue avant de me livrer la vidéo ; aucun chiffre à l'écran sans une source vérifiée que tu me cites.
```

Ce prompt ne faisait pas partie du test : on ne l'a pas lancé. La doc dit que Claude Code voit les nouvelles skills en cours de session, sauf si le dossier .claude/skills n'existait pas au départ. Dans ce cas, tape /reload-skills ou ouvre une nouvelle session.

D'où vient chaque règle :

• 1,5 s : notre remarque de l'étape 6. Claude en a fait un contrôle automatique dans `voix.py`, qui affiche « TROP LONG ».

• L'heure du mot : l'option WordBoundary d'edge-tts, trouvée par Claude à l'étape 6. En v1, sans elle, le timing des sous-titres était estimé.

• 3 mots au plus : le découpage des sous-titres de la v2 (réglé à 3 dans `voix.py`).

• Le silence coupé : en v2, chaque phrase s'arrête 0,2 s après son dernier mot. C'est ce qui fait passer la vidéo de 43 à 37 s.

• −14 LUFS : Claude l'a appliqué seul à l'étape 5, après avoir mesuré −25,7 dB de moyenne.

• La planche d'une image par seconde : Claude l'a faite à l'étape 6 et y a trouvé trois défauts. C'est aussi sur notre planche de la v1 qu'on a vu la suite de diapos.

• Les chiffres sourcés : à l'étape 5, Claude écrit lui-même « Je les ai écrits de mémoire sans les revérifier ; vérifie-les avant de publier. »

9. Les limites

La voix

edge-tts passe par le service de synthèse vocale en ligne de Microsoft Edge, d'après son README. C'est gratuit et sans clé, mais rien ne tourne en local, alors que Claude avait repéré les voix françaises de macOS (Thomas, Amélie, Jacques). On n'a pas jugé la voix à l'oreille.

Les chiffres du reel d'exemple

Les –30 %, 85 dB, 2 000 m et moins de 20 % d'humidité ont été écrits de mémoire. Claude le signale aux étapes 5 et 6, et personne ne les a vérifiés depuis. Ne les reprends pas tels quels.

Les visuels

La v2 tient le rythme demandé. Elle reste simple : des emoji Apple et du texte sur un dégradé, sans musique. On est loin du niveau de détail des reels animés qu'on voit passer sur Instagram.

L'appli desktop

On ne pouvait pas piloter l'appli depuis notre session. Le test a donc été joué avec Claude Code en ligne de commande (version 2.1.280), lancé dans le dossier, avec le même modèle et le même abonnement. La doc de Claude Code dit que l'appli, le terminal et les autres versions passent par le même moteur. Pour partir de zéro, comme un nouvel utilisateur, on avait coupé nos réglages perso, notre mémoire et nos skills (option --safe-mode).

Le reste

Au milieu des étapes 5 et 6, ses messages de travail passent en anglais. Ses comptes rendus de fin restent en français.

Chaque prompt a été lancé une seule fois. Chez toi, le script, les outils choisis et les durées peuvent changer.

10. Sources

Relevés du 2 novembre 2026, heure de Paris.

Notre test

• Traces brutes des trois tours (début et fin à la seconde, chaque commande, chaque réponse), une seule session de 13 h 36 à 13 h 52 : Claude Code 2.1.280 en ligne de commande, modèle claude-opus-5-5, abonnement Claude, aucune clé d'API.

• Versions relevées par Claude : Node 22.22.2, ffmpeg 8.0.1, Python 3.14.5, uv 0.9.26, ImageMagick 7.1.2-18, macOS 26.6.2. Installées dans le dossier : Remotion 4.0.540, React 19.3.1, TypeScript 7.0.2, edge-tts 7.3.1.

• Durées et coût équivalent API, tirés des traces : 5 min 35 et 0,33 dollar, 2 min 22 et 0,63 dollar, 7 min 21 et 1,87 dollar.

• Nos contrôles du piège du dossier : copie « avec espace » et copie au même chemin hors du Bureau, juste après le test ; copie dans Documents, chargement et rendu de 2 s dans le dossier d'origine, entre 14 h 34 et 14 h 40.

Claude

• Page de téléchargement de Claude (https://claude.com/download), capturée à 14 h 22. claude.ai/download y redirige.

• Présentation de Claude Code (https://code.claude.com/docs/en/overview) : installeurs de l'appli, abonnement payant, et la phrase « Each surface connects to the same underlying Claude Code engine ».

• Claude Code dans l'appli desktop (https://code.claude.com/docs/en/desktop) : démarrer une session, environnements Local, Cloud et SSH, choix du dossier.

• Skills de Claude Code (https://code.claude.com/docs/en/skills) : emplacement des skills de projet, rôle de la description, prise en compte en cours de session, /reload-skills.

• Textes français de l'appli Claude 4.4212.0, lus dans son paquet sur notre Mac : « Environnement », « Local », « Sélectionner un dossier ».

Les outils

• Licence de Remotion (https://github.com/remotion-dev/remotion/blob/main/LICENSE.md) : gratuite pour un particulier et pour une entreprise de 3 salariés au plus.

• edge-tts (https://github.com/rany2/edge-tts) : « a Python module that allows you to use Microsoft Edge's online text-to-speech service ».

Non vérifié

• La cause du blocage de Remotion pendant le test.

• Le même parcours cliqué dans l'appli desktop.

• Le comportement du prompt d'installation sur un Mac sans Node ni ffmpeg.

• Le prompt de la skill de la partie 8.

• La voix à l'oreille, et les chiffres du reel d'exemple.
