Eigenflow

GUIDE · 11 NOVEMBRE 2026

Blender Skills pour Claude Code

Installer le pack sur Mac et Windows, éviter le piège du .mcp.json, et ce que Claude a vraiment rendu dans notre test

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté BLENDER, voici le guide. Blender Skills est un pack gratuit de 94 skills qui apprend à Claude Code (ou à Cursor, Kiro et Codex) à travailler dans Blender comme une équipe de production 3D : il planifie, découpe le travail, fixe des budgets de polygones, nomme les objets proprement et prépare l'export vers un moteur de jeu. Pour que Claude agisse vraiment dans Blender, il faut une deuxième brique, l'extension MCP for Blender.

Tout a été relu le 11 novembre 2026 : les deux dépôts, le code de l'extension, la documentation de Claude Code, la page de prix de Claude vue de France, les licences. On n'a rien installé. Les commandes viennent des README officiels et de la documentation, et on te signale celles qu'on a adaptées.

On n'a aucun lien avec l'auteur du pack, ni avec celui de l'extension, ni avec Anthropic ou la Blender Foundation.

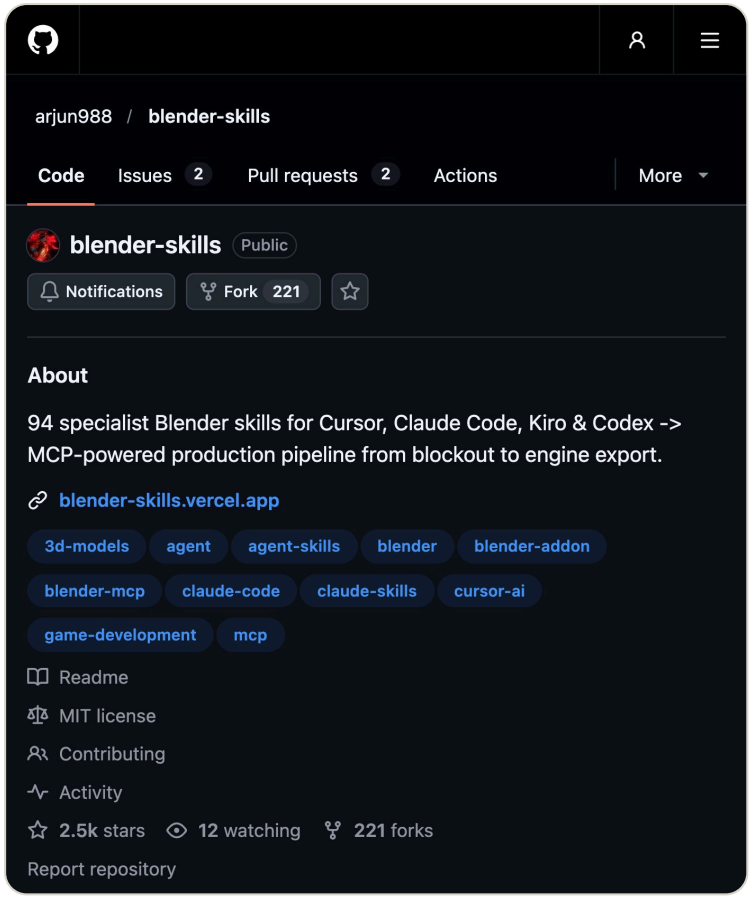
À savoir avant de lire

- Le pack est un dossier de texte : 94 skills, 269 fichiers Markdown, aucun programme. Seul, il ne touche pas à Blender. C'est l'extension MCP for Blender qui donne la main à Claude, en lui faisant exécuter du Python dans ton fichier ouvert.
- Le pack (licence MIT), Blender (GPL) et l'extension (MIT) sont gratuits. Claude Code est payant : au minimum la formule Pro, 18 € HT par mois sans engagement en France, soit 21,60 € TTC, ou 15 € HT par mois à l'année, au 11 novembre 2026.
- Le fichier .mcp.json du dépôt est écrit pour Windows. Sur Mac et Linux, il empêche l'extension de démarrer. Un utilisateur l’a signalé le 10 novembre 2026. Les parties 3 à 5 te donnent la bonne commande.
- L'extension exécute le code que Claude écrit, sans limite par défaut. Enregistre ton fichier avant chaque session et active son mode sûr (partie 6).
- Par défaut, l'extension envoie un relevé d'usage anonyme à son auteur. Une variable le coupe (partie 6).
- 63 des 94 skills font moins de 3 Ko : quelques lignes de direction artistique. Le gros du savoir-faire tient dans 31 skills, la director en tête (partie 7).
- Dans notre test, sans Blender branché, Claude a rendu un plan de production et un script Python de 353 lignes, jamais lancé. Il l'a écrit lui-même : un monde entier, c'est des mois de production.

1. Ce que c'est, en vrai

Blender Skills est un dépôt GitHub, arjun988/blender-skills, sous licence MIT. Au relevé du 11 novembre 2026 : 2 478 étoiles, 221 copies (forks), version 1.3.0, un seul contributeur. Le dépôt est

né le 5 juillet 2026. Ses skills n'ont plus changé depuis le 10 juillet ; le dernier envoi, le 6 novembre, ajoute un graphique d'étoiles au README.



La page GitHub du pack Blender Skills le 11 novembre 2026 : 94 skills pour Cursor, Claude Code, Kiro et Codex, licence MIT, 2 500 étoiles arrondies et 221 forks

Une skill, c'est un dossier avec un fichier SKILL.md : un en-tête (nom, description qui dit quand s'en servir, licence) puis des consignes en anglais, parfois des fiches de référence à côté. Le pack compte 94 skills, plus 9 fiches partagées (chaîne de production, nommage, budgets de polygones, listes de contrôle). En tout, 175 fiches, et pas une ligne de code à exécuter.

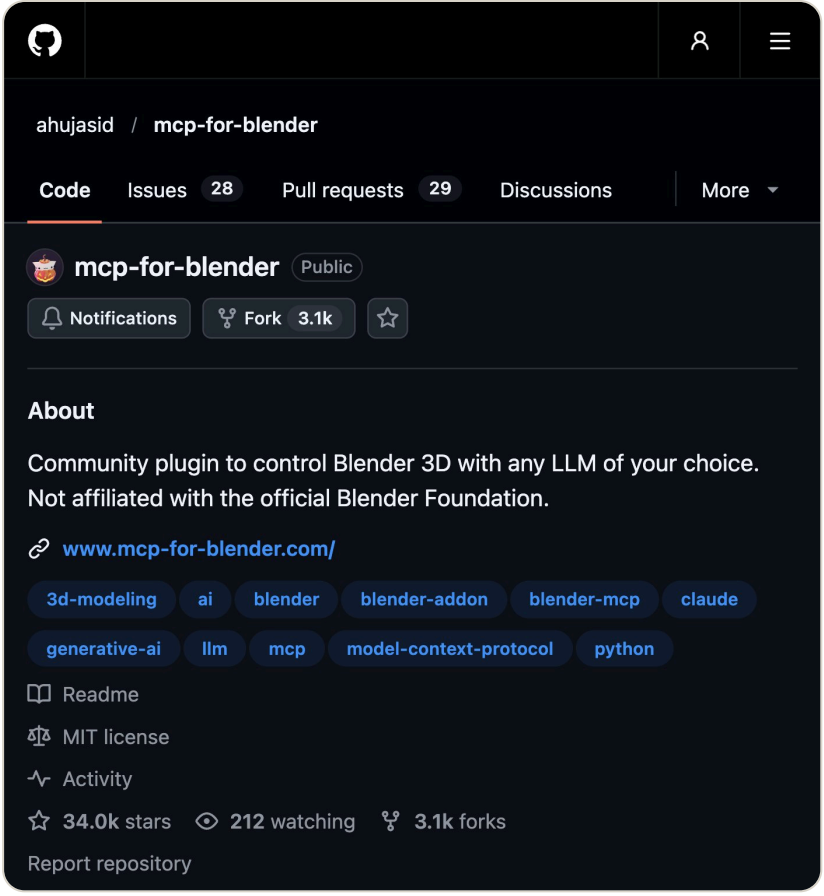
Claude Code lit le nom et la description de chaque skill installée à chaque échange, et charge le texte complet quand ta demande correspond. Tu peux aussi en appeler une toi-même : /blender-director.

La chaîne complète a quatre maillons :

- Claude Code lit les skills et écrit du Python pour Blender.
- Le serveur de l'extension, que Claude Code lance avec uv, transmet ce code.
- Le module complémentaire de l'extension, installé dans Blender, écoute sur ton ordinateur (port 9876) et exécute le code.
- Blender, ouvert, avec ton fichier.

Ce que fait l'extension

MCP for Blender (ahujasid/mcp-for-blender, anciennement blender-mcp) est un projet indépendant sous licence MIT, avec environ 34 000 étoiles le 11 novembre 2026. Sa description GitHub le précise : « Not affiliated with the official Blender Foundation. » Elle existe depuis mars 2025, seize mois avant le pack.



La page GitHub de l'extension MCP for Blender le 11 novembre 2026 : « Community plugin to control Blender 3D with any LLM of your choice. Not affiliated with the official Blender Foundation. », licence MIT, 34 000 étoiles

Elle donne neuf outils à Claude. Ceux qui comptent :

- execute_blender_code exécute du Python dans ton Blender ouvert. Modélisation, matériaux, lumières, caméras, animation, export : tout passe par là.
- look renvoie une image du viewport, de la caméra, en filaire ou sous plusieurs angles, pour que Claude voie ce qu'il a fait.
- get_scene_info résume la scène, un objet par ligne.
- search_assets et import_asset vont chercher des HDRI, textures et modèles sur Poly Haven, et des modèles sur Sketchfab et Poly Pizza.
- generate_3d génère un modèle par IA (Meshy, Tripo, Hunyuan3D, Hyper3D Rodin), avec tes propres clés API ou par l'offre Premium payante.
- get_addon_status, disable_telemetry et record_trajectory_feedback servent à l'état du module et à la collecte de données.

Le pack a été écrit pour une version plus ancienne de l'extension : il cite get_viewport_screenshot, devenu look. Il prévient lui-même que la liste fournie par l'extension fait foi, et Claude la lit à la connexion.

---

2. Ce qu'il te faut

Blender

La version 5.3.0, sortie le 10 novembre 2026, gratuite, sur blender.org. L'extension accepte Blender 3.0 et plus ; le script de notre test vise Blender 4.2 et plus.



La page de téléchargement de blender.org le 11 novembre 2026 : Blender 5.3.0 pour macOS Apple Silicon, 343 Mo

Claude Code, ou un autre agent

Claude Code est inclus dans toutes les formules payantes de Claude, pas dans la gratuite. Page de prix relevée depuis la France le 11 novembre 2026 :

- Free : 0 €, sans Claude Code.
- Pro : 18 € HT par mois sans engagement, ou 15 € HT par mois à l'année (180 € facturés d'avance).
- Max : 90 € HT par mois.
- Ultra : 180 € HT par mois.

La page précise : « Les prix affichés n'incluent pas les taxes applicables. » Avec la TVA française à 20 %, Pro revient à 21,60 € TTC par mois. Claude Code puise dans le même quota que tes conversations. La même page cite une autre voie : des crédits API à la demande, par un compte Console.



La carte Pro sur la page de prix de Claude vue de France le 11 novembre 2026 : 15 € par mois à l'année, 18 € si facturé mensuellement

Le README du pack annonce aussi Cursor, Kiro et Codex. Leurs prix n'ont pas été vérifiés pour ce guide, qui détaille Claude Code.

uv et git

uv lance le serveur de l'extension et télécharge lui-même un Python qui convient (3.10 ou plus). git sert à récupérer le pack ; sans git, le bouton Code puis Download ZIP de la page GitHub fait la même chose.

---

3. Installer sur Mac, pas à pas

Toutes les commandes de cette partie se tapent dans le Terminal, pas dans Claude.

Étape 1 : Blender

Télécharge Blender 5.3 sur blender.org, glisse-le dans Applications et ouvre-le une première fois.

Étape 2 : uv

Si tu as Homebrew :

```
À COLLER
brew install uv
```

Sinon, l'installateur officiel d'uv :

```
À COLLER

curl -LsSf https://astral.sh/uv/install.sh | sh
```

Ferme le Terminal, rouvre-le, puis vérifie que la commande répond :

```
À COLLER

uvx --version
```

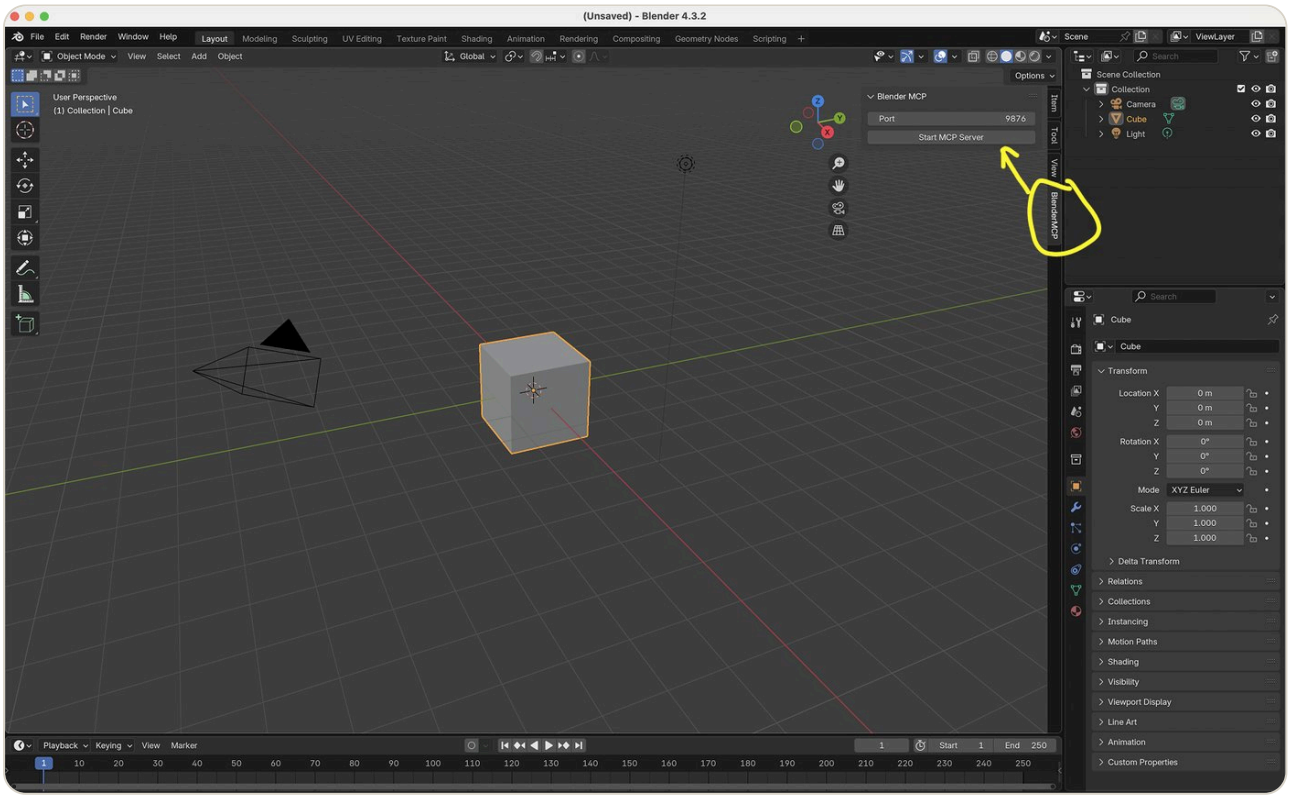
Le README de l'extension insiste : n'installe pas uv avec pip install uv, la commande uvx risque de manquer.

Étape 3 : le module dans Blender

```
À COLLER

uvx mcp-for-blender install-addon
```

La commande copie le module dans le dossier des add-ons de Blender. Dans Blender, ouvre Edit, Preferences, Add-ons, cherche « MCP for Blender » et coche Interface: MCP for Blender (redémarre Blender s'il n'apparaît pas). Dans la vue 3D, appuie sur N, onglet MCP for Blender : le serveur démarre seul, sinon clique sur Start MCP Server. Port par défaut : 9876.



Le panneau de l'extension dans la vue 3D de Blender, avec le port 9876 et le bouton Start MCP Server. Image tirée du README de l'extension, sur une version plus ancienne où l'onglet s'appelait encore Blender MCP

Étape 4 : les 94 skills dans un dossier de projet

Crée un dossier réservé à ton travail dans Blender, et copies-y les skills :

```
À COLLER

cd ~
git clone https://github.com/arjun988/blender-skills.git
mkdir -p ~/blender-claude/.claude/skills
cp -R ~/blender-skills/.claude/skills/. ~/blender-claude/.claude/skills/
```

Notre commande diffère de celle du README sur un point. Le README écrit cp -r blender-skills/.claude/skills /path/to/your/project/.claude/skills/ : si .claude/skills existe déjà dans ton projet, ça crée un dossier skills dans skills, et Claude Code ne voit plus rien. Le /. copie le contenu, pas le dossier. Et on ne copie que les skills, surtout pas le .mcp.json du dépôt (partie 5).

Pourquoi un dossier à part plutôt que ~/.claude/skills ? Chaque skill installée ajoute sa description au contexte à chaque échange, 3D ou pas, et la documentation de Claude Code réserve à cette liste 1 % de la fenêtre de contexte. Dans un dossier de projet, les 94 ne se chargent que là.

Étape 5 : relier Claude Code à l'extension

Place-toi dans le dossier du projet, puis tape la commande officielle, celle du README de l'extension :

```
À COLLER

cd ~/blender-claude
claude mcp add blender uvx mcp-for-blender
```

Notre version fait la même chose, avec la télémétrie coupée et le mode sûr activé (partie 6). On l'a écrite d'après la documentation de Claude Code, sans la tester :

```
À COLLER

cd ~/blender-claude
claude mcp add --env DISABLE_TELEMETRY=true \
  --env BLENDER_MCP_SAFE_MODE=1 \
  --transport stdio blender -- uvx mcp-for-blender
```

Les barres obliques inverses en fin de ligne servent juste à couper la commande en trois. Le --transport stdio est placé exprès entre les --env et le nom : la documentation prévient que si le nom suit directement un --env, Claude Code le lit comme une variable de plus. Le -- sépare les options de Claude Code de la commande du serveur. Par défaut, le serveur n'est enregistré que pour ce dossier ; ajoute --scope user après add pour l'avoir partout.

Vérifie :

```
À COLLER

claude mcp list
```

Puis lance claude dans le dossier et tape /mcp : blender doit apparaître connecté, Blender ouvert et serveur démarré. Le README de l'extension ajoute trois consignes : ne lance pas toi-même uvx mcp-for-blender, Claude Code s'en charge ; un seul client à la fois (Claude Code, Claude Desktop ou Cursor) ; le premier appel échoue parfois, le suivant passe.

L'installateur en une ligne, si tu préfères

Le README de l'extension propose un script qui fait les étapes 2, 3 et 5 d'un coup : il cherche tes applis IA, ajoute une entrée blender à celles que tu choisis (avec une copie .bak), puis installe et active le module. Ne le lance pas sans l'avoir lu :

```
À COLLER

curl -LsSf https://www.mcp-for-blender.com/install.sh -o install.sh
less install.sh
sh install.sh
```

On l'a lu le 11 novembre sans le lancer : 61 lignes, il installe uv s'il manque, puis lance uvx mcp-for-blender@latest setup. Il ne coupe pas la télémétrie.

---

4. Installer sur Windows

Les commandes de cette partie se tapent dans PowerShell.

Étapes 1 à 3 : Blender, uv, le module

Installe Blender 5.3 depuis blender.org. Puis uv, avec la commande que donne l'extension :

```
À COLLER

winget install --id=astral-sh.uv -e
```

Ouvre une nouvelle fenêtre PowerShell, puis installe le module :

```
À COLLER

uvx mcp-for-blender install-addon
```

Active-le dans Blender (Edit, Preferences, Add-ons, Interface: MCP for Blender), puis appuie sur N dans la vue 3D et ouvre l'onglet MCP for Blender. Le README conseille cette voie winget si Windows Defender bloque l'installateur en une ligne.

Étape 4 : les skills

Le README du pack donne la commande pour Cursor sous Windows. Pour Claude Code, on l'a adaptée au dossier .claude :

```
À COLLER

cd $HOME
git clone https://github.com/arjun988/blender-skills.git
$dest = "$HOME\blender-claude\.claude\skills"
New-Item -ItemType Directory -Force $dest
Copy-Item -Recurse -Force "$HOME\blender-skills\.claude\skills\*" $dest
```

Étape 5 : relier Claude Code

```
À COLLER

cd $HOME\blender-claude
claude mcp add blender uvx mcp-for-blender
```

Si claude mcp list montre que blender ne se connecte pas, retire-le et ajoute-le avec l'enveloppe cmd /c, celle que le README de l'extension donne pour Windows. Adaptée à Claude Code par nos soins, non testée :

```
À COLLER

claude mcp remove blender
claude mcp add --transport stdio blender -- cmd /c uvx mcp-for-blender
```

Autre solution du README : remplacer uvx par son chemin complet, que la commande where uvx affiche.

---

5. Le piège du fichier .mcp.json

Le dépôt du pack contient un fichier .mcp.json, censé relier Claude Code à Blender tout seul. Le voici tel qu'il est le 11 novembre 2026 :

À COLLER

```
{
  "mcpServers": {
    "blender": {
      "command": "cmd",
      "args": ["/c", "uvx", "--python", "3.11", "blender-mcp"],
      "env": {
        "BLENDER_HOST": "localhost",
        "BLENDER_PORT": "9876",
        "UV_PYTHON_PREFERENCE": "only-managed",
        "DISABLE_TELEMETRY": "true"
      }
    }
  }
}
```

cmd est l'invite de commandes de Windows. Sur Mac et Linux, ce programme n'existe pas : le serveur ne démarre jamais, et Claude ne voit aucun outil Blender. Un utilisateur l'a signalé le 10 novembre 2026 dans le ticket #4 du dépôt, « Blender MCP server does not start on macOS/Linux », avec Claude Code 2.1.296. Pas de réponse au 11 novembre.

Le piège se déclenche de trois façons :

- tu copies tout le dépôt dans ton projet, ou ce fichier à sa racine ;
- tu lances Claude Code dans le dossier cloné ;
- tu installes le pack comme plugin. La documentation de Claude Code l'écrit : « Claude Code loads .mcp.json at the plugin root first ». La racine du plugin est celle du dépôt, donc ce fichier part avec. Sur Windows, ce serveur démarre, et si tu as aussi ajouté le tien, tu en as deux, alors que l'extension demande de n'en faire tourner qu'un.

D'où notre conseil : copier le seul dossier des skills (étape 4) et passer par claude mcp add (étape 5). Tu gardes la main sur la ligne qui lance le serveur.

Trois autres erreurs dans la documentation du pack :

- Le QUICKSTART propose /plugin install blender-skills@arjun988. Le fichier marketplace.json du dépôt nomme la marketplace blender-skills. La commande qui correspond serait blender-skills@blender-skills, celle qu'emploie l'auteur du ticket #4.
- Le guide d'installation du pack donne claude mcp add blender cmd /c uvx --python 3.11 blender-mcp. Il manque -- avant cmd : sans lui, d'après la documentation, Claude Code essaie de lire --python comme une de ses propres options. Et la ligne ne vaut que pour Windows.
- Le README annonce « Marketplace listing is currently under review by Anthropic. » Impossible à vérifier. Le 11 novembre, le pack ne figure pas dans la marketplace officielle d'Anthropic : 317 plugins, aucun sur Blender.

Si tu as déjà installé le plugin sur Mac, le ticket #4 donne un contournement : remplacer cmd et /c par uvx dans la copie du fichier que Claude Code garde en cache, à refaire après chaque mise à jour. Plus simple, désinstalle le plugin et passe par la copie du dossier :

À COLLER

```
claude plugin uninstall blender-skills@blender-skills
```

---

6. Sécurité : ce que tu autorises

Du Python sans limite

L'outil execute_blender_code exécute n'importe quel code Python dans Blender, avec tes droits d'utilisateur : lire et écrire des fichiers, lancer des programmes, aller sur internet. Le README de l'extension l'écrit en gras : « ALWAYS save your work before using it. » Ses conditions d'utilisation ajoutent : « You are responsible for reviewing any AI-generated code before using it. »

D'où trois habitudes. Enregistre ton .blend avant chaque session et travaille sur une copie de tout fichier qui compte : le script de notre test créait une nouvelle scène sans toucher aux autres, rien ne le garantit pour le suivant. Lis ce que Claude s'apprête à exécuter avant d'accepter. Et garde le serveur du module sur ton ordinateur : il n'a « no authentication or encryption », dit le README, donc jamais l'option Allow Remote Connections sur un réseau que tu ne maîtrises pas.

Le mode sûr

Avec BLENDER_MCP_SAFE_MODE=1, présent dans notre commande de l'étape 5, l'extension vérifie chaque script avant de l'exécuter. Son code, lu le 11 novembre, n'autorise que bpy, bmesh, mathutils et une vingtaine de modules sans accès aux fichiers ni au réseau (math, random, json...). Il bloque eval, exec, open, os, subprocess, le réseau et le chargement de fichiers .blend extérieurs. Rendu, enregistrement, import et export restent permis. Un script refusé revient à Claude avec la raison.

Le revers : le script de notre test importe os pour créer son dossier d'export. En mode sûr, il serait refusé tel quel. Une phrase suffit à le faire réécrire (partie 10).

La télémétrie

Sans réglage, l'extension envoie à son auteur un relevé anonyme à chaque usage : identifiant d'installation tiré au hasard, session, nom de l'outil, réussite, durée, versions, système, heure.

Tes prompts, ton code et tes captures ne partent que si tu coches le consentement dans les préférences du module. Ses conditions, mises à jour en août 2026, donnent alors à l'auteur une licence « worldwide, royalty-free, perpetual » sur ces données, pour entraîner des modèles et publier des jeux de données, et prévoient d'enregistrer aussi tes opérations faites à la main dans Blender. Certaines applis proposent ce consentement en début de conversation : tu peux refuser.

Pour tout couper, relevé anonyme compris : DISABLE_TELEMETRY=true dans l'environnement du serveur, comme dans notre commande de l'étape 5. L'outil disable_telemetry ne coupe que la collecte du contenu.

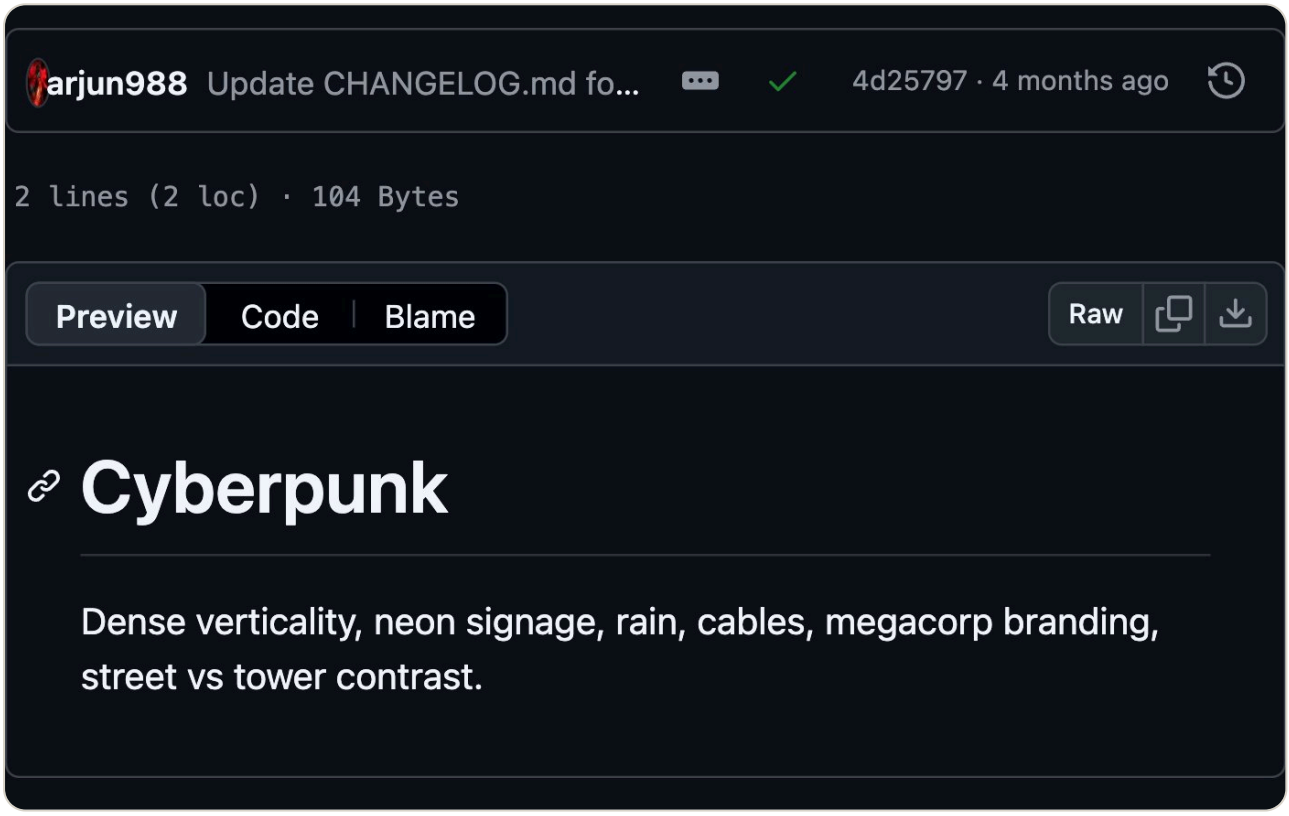
Côté pack, on a vérifié : 269 fichiers Markdown, sans script, sans hook. Les risques viennent de l'extension et du fichier .mcp.json.

7. Les 94 skills : comment elles sont faites

Les 63 courtes

Rangées par taille de dossier, 63 skills font moins de 3 Ko, avec une médiane de 948 octets, soit une quinzaine de lignes. 51 d'entre elles sont des styles (anime, pixel art, noir, voxel), des genres de jeu (tir, RPG, soulslike), des mondes et des ambiances (fantasy, cyberpunk, cozy) ou des packs horreur. Les 12 autres sont des étapes techniques : exports Unity, Unreal et Godot, LOD, collisions, contrôle qualité, assemblage de scène, habillage, archviz, lookdev, végétation, archétypes de personnages.

L'exemple de notre test : la fiche cyberpunk tient en deux lignes et 104 octets. Le SKILL.md de sci-fi-punk-worlds, qui la contient, fait 804 octets.



La fiche cyberpunk du pack sur GitHub : 2 lignes, 104 octets, « Dense verticality, neon signage, rain, cables, megacorp branding, street vs tower contrast. »

Claude connaît déjà la plupart de ces mots-clés. Ce que ces fiches apportent : un vocabulaire commun pour le routage, et quelques règles de direction, comme « Don't mix chrome cyber with brass steam unless intentional mashup ». Personne n'a publié de comparaison avec et sans le pack. La question est posée dans le ticket #2 du dépôt depuis le 25 octobre, sans réponse au 11 novembre.

Les 31 plus fournies

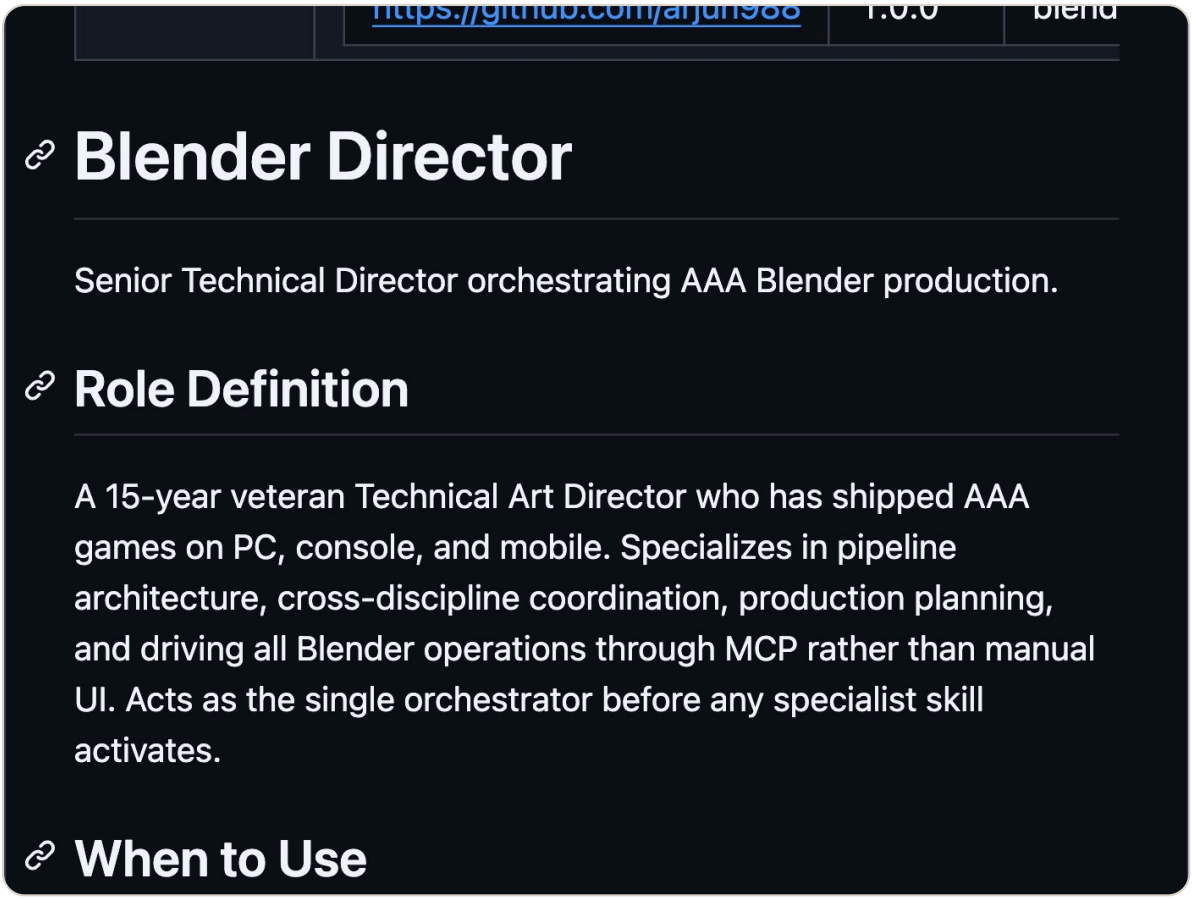
Taille du dossier complet de chaque skill (SKILL.md et fiches de référence), en kilo-octets de texte :

- blender-director, 23,9 Ko : le chef d'orchestre (partie 8).
- blender-modeler, 7,5 Ko : modélisation générale, modificateurs, collections.
- hard-surface, 7,4 Ko : science-fiction, armes, machines, booléens et chanfreins.

- environment-artist, 7,2 Ko : kits modulaires, bâtiments, terrain.
- realistic-style, 7,0 Ko : photoréalisme, échelle réelle, PBR.
- horror-style, 7,0 Ko : horreur psychologique.
- rendering, 6,6 Ko : Cycles et EEVEE, échantillons, passes de rendu.
- stylized-style, 6,6 Ko : formes, exagération, rendu non photoréaliste.
- creature-artist, 6,5 Ko : monstres, créatures, anatomie.
- texture-workflow, 6,1 Ko : atlas, cuisson des maps, décalcomanies.
- retopology, 6,0 Ko : maillage propre pour l'animation.
- rigging, 6,0 Ko : armatures, IK et FK, poids.
- lighting, 5,9 Ko : éclairage cinéma et jeu, HDRI, volumétrie.
- character-artist, 5,8 Ko : personnages humains, topologie du visage.
- uv-workflow, 5,7 Ko : coutures, densité de texels, UDIM.
- geometry-nodes, 5,6 Ko : procédural, dispersion d'objets.
- materials, 5,5 Ko : matériaux PBR et stylisés.
- sculpting, 5,4 Ko : sculpture, détails organiques.
- animation, 5,2 Ko : cycles de marche, combat, NLA.
- lowpoly-style, 5,0 Ko : style PS1 et PS2.
- export-pipeline, 4,9 Ko : FBX, glTF, USD, Alembic.
- prop-artist, 4,4 Ko : objets du quotidien, meubles.
- asset-optimization, 4,3 Ko : polygones, topologie et nommage avant livraison.
- vehicle-artist, 4,2 Ko : voitures, avions, bateaux, mechas.
- hair-groom, 3,6 Ko : cheveux et fourrure.
- procedural-modeling, 3,6 Ko : rochers, routes et câbles en procédural.
- cloth-sim, 3,5 Ko : vêtements, simulation de tissu.
- camera-cinematography, 3,5 Ko : cadrage, objectifs, mouvements de caméra.
- vfx-fx, 3,2 Ko : fumée, feu, particules.
- physics-sim, 3,2 Ko : corps rigides, destruction.
- compositing, 3,0 Ko : compositeur, étalonnage.

S'y ajoutent les 9 fiches partagées (35 Ko) que toutes les skills citent : chaîne de production, budgets de polygones, nommage (préfixes COL_, SM_, MAT_, LGT_, CAM_), listes de contrôle avant livraison.

8. La skill blender-director



Le début de la skill blender-director sur GitHub : « Senior Technical Director orchestrating AAA Blender production »

Elle se présente en directeur artistique technique avec quinze ans de jeux AAA, qui passe avant toute autre skill. Sa méthode, en sept étapes :

1. Analyser : type d'objet, plateforme cible, direction artistique.

2. Planifier : choisir les skills, fixer les budgets de polygones et de textures, le format d'export.

3. Écrire le brief de production, avant toute opération dans Blender.

4. Déléguer aux skills spécialisées, dans l'ordre de la chaîne.

5. Exécuter par l'extension.

6. Valider avec la liste de contrôle.

7. Livrer : l'export et un rapport de validation.

Avant de commencer, elle tranche six questions : le rôle de l'objet (héros, standard ou fond), la cible (jeu, cinéma, portfolio), la plateforme, le style, l'animation et le format d'export.

Le Production Brief a toujours la même forme : type, cible, style, budget de polygones, budget de textures, export, pivot, animation, skills actives, phases à cocher et plan d'exécution.

Le routage tient dans un tableau de 91 lignes : un signal dans ta demande, une skill principale, des skills d'appui. Pour « un monde cyberpunk pour Unreal Engine », il mène à sci-fi-punk-worlds pour le cyberpunk, neon-retrofuturism pour les néons, environment-artist pour les bâtiments et les kits modulaires, et unreal-export, avec export-pipeline, lod-pipeline et collision-proxy en appui.

Ses règles, en résumé : le brief avant de modéliser, des collections préfixées COL_ dès le départ, des budgets écrits noir sur blanc, jamais « terminé » sans validation, jamais d'explication de l'interface quand l'extension peut agir, jamais d'opération destructive sans sauvegarde.

Quand l'extension n'est pas branchée, la fiche partagée mcp-integration.md prévoit un mode de repli : un plan concis et des scripts Python « the user can run ». C'est ce qui s'est passé dans notre test.

---

9. Notre vrai test du 11 novembre 2026

Le montage

On n'a installé ni Blender, ni l'extension, ni le pack. On a donné à Claude Opus 5.5, en consigne système, dix fichiers du pack : la director et son tableau de routage, quatre fiches partagées et les quatre skills qu'elle route pour cette demande (sci-fi-punk-worlds, neon-retrofuturism, environment-artist, unreal-export). Lancé avec claude -p, sans aucun outil, dans un dossier vide : une session où la skill est chargée, sans Blender branché.

La demande, celle de la vidéo :

> **À COLLER**
>
> `Construis-moi un monde cyberpunk pour Unreal Engine.`

Réponse en 7 minutes 16 secondes. Le compteur indiquait 1,12 $ au prix de l'API ; rien n'a été facturé, la session passait par l'abonnement.

Ce que Claude a répondu d'abord

Deux phrases posent le cadre : « Aucun outil Blender MCP n'est connecté à cette session. Je ne peux donc pas piloter Blender, vérifier la scène ni comparer des captures d'écran. » Puis : « Un monde entier représente des mois de production. Je commence par la base dont tout le reste dépend : un kit modulaire et une rue assemblée, exportables vers UE5. »

Le brief

- Cible : Unreal Engine 5 avec Lumen, PC et console ; un kit modulaire et une rue assemblée.
- Style : cyberpunk seul, néons magenta, cyan et ambre, brume volumétrique.
- Budgets : mur de 4 m entre 200 et 800 triangles, section d'immeuble héros entre 5 000 et 20 000, LOD à 50, 25 et 10 %. Le pack donne pourtant 200 à 800 pour un mur de 2 m.
- Textures : une trim sheet de 4096 partagée, matériaux répétables en 2048.
- Export : un FBX par module, axes -Z Forward et Y Up, collisions UCX_, grille de 4 m.
- Skills actives : sci-fi-punk-worlds, neon-retrofuturism, environment-artist, collision-proxy, unreal-export, lod-pipeline. Les deux dernières, il les a tirées du tableau de routage sans les avoir sous les yeux.
- Kit : trois murs (plein, fenêtre, rideau de commerce), un pilier, un toit, une voie, un trottoir, un climatiseur, trois enseignes néon et une tour de 215 m pour le fond.

Standard NPC	20,000–40,000	Background characters
Crowd	5,000–10,000	Distant crowds, impostors
Mobile hero	15,000–30,000	Mobile/ Switch targets

Environment

Asset	Triangles	Notes
Modular wall (2m)	200–800	Repeats frequently
Modular floor tile	100–400	Instanced
Hero building section	5,000–20,000	Landmark structures
Rock (hero)	1,000–5,000	Scatter dependent
Tree (realistic)	5,000–15,000	Use cards/impostors at distance
Tree		

La fiche des budgets du pack : 200 à 800 triangles pour un mur modulaire de 2 m, 5 000 à 20 000 pour une section d'immeuble héros

Le script

353 lignes de Python pour Blender. Il crée une nouvelle scène, SCN_Cyberpunk_District, sans toucher aux autres, sept collections, douze matériaux, les douze modules du kit en blocs et leurs collisions. Puis une rue de 64 m tirée au hasard avec une graine fixe, 2077 : 12 immeubles de 3 à 10 étages, 22 climatiseurs, 11 enseignes néon, 9 tours au loin, une caméra à hauteur d'œil, une lune, une lumière sous chaque enseigne, de la brume. L'export FBX est écrit mais coupé : il faut passer RUN_EXPORT à True.

On ne l'a pas lancé dans Blender. Pour la vidéo, on a rejoué son assemblage en Python, mêmes graine et tirages : 442 pièces, dont 433 dans la rue, pour 22 428 triangles. Un module fait de 12 triangles (le pilier) à 96 (le mur à fenêtre).



La rue du script de Claude, rejouée pièce par pièce dans notre moteur pour la vidéo : des blocs, sans texture

Ce qu'il couvre, ce qu'il laisse

Claude a fait la liste lui-même. Le script couvre le planning, la silhouette et les ruptures de panneaux (niveaux 1 et 2 sur 3), des UV provisoires, des matériaux de base, les collisions et l'export. Restent les détails (câbles, tuyaux, hologrammes), la trim sheet, les LOD, la validation dans Unreal et les angles de rue.

Ce qu'il n'y a pas :

- aucun personnage, seulement un mannequin d'échelle, une boîte orange de 1,80 m ;
- aucune texture image, des matériaux unis ;
- aucune animation ;
- aucune optimisation. Les modules sont légers parce que c'est un blockout, et Claude le dit : « C'est voulu : le budget se dépense au niveau 3. »

Et sa phrase la plus utile : « Le script n'a pas été exécuté ni testé ici. Il pourrait donc contenir une erreur que je n'ai pas vue. Si Blender en signale une, colle-moi le message. »

Les questions qu'il a posées

- Le monde sera-t-il jouable, en vue subjective ou à la troisième personne, ou cinématique ? Ça change la largeur des rues, les collisions et l'accès aux toits.
- Quelle version d'Unreal ? Avec Nanite, il assouplirait les budgets des objets statiques.
- Quelle référence visuelle : Blade Runner, Cyberpunk 2077, ou plus stylisé ? Une image l'aiderait.
- Et une proposition : brancher l'extension, pour qu'il exécute le script lui-même et corrige à partir de captures. Il renvoie vers le .mcp.json du pack, celui qui ne marche pas sur Mac.

10. Premières consignes à coller

Dans Claude Code, lancé dans ton dossier de projet, avec Blender ouvert et le serveur démarré. Une consigne par étape, dans cet ordre.

Vérifier la connexion, sans rien toucher :

À COLLER

```
Avec l'outil get_scene_info, décris la scène Blender ouverte : objets,
collections, matériaux. Ne modifie rien.
```

Un premier objet, petit et mesurable :

À COLLER

```
/blender-director Un tonneau en bois pour un jeu low poly, 800 triangles
maximum, export glTF. Donne-moi d'abord le brief et attends mon accord. Ensuite
construis-le dans Blender par petites étapes, montre-moi le résultat avec
l'outil look après chaque étape, et n'exporte rien sans me demander.
```

La version réaliste du monde cyberpunk, un module à la fois :

À COLLER

```
/blender-director Un seul module de façade cyberpunk de 4 m pour Unreal Engine 5
: mur, fenêtre et une enseigne néon, dans le budget du pack pour un mur
modulaire. Enregistre une copie du fichier .blend avant de commencer. Montre-moi
le module en filaire avec l'outil look, puis donne-moi son nombre de triangles.
```

La lumière, avec Poly Haven (coche d'abord Poly Haven dans le panneau de l'extension) :

À COLLER

```
Éclaire la scène avec un HDRI de nuit en ville trouvé sur Poly Haven, en 2k.
Donne-moi son nom et le lien de sa page.
```

Le contrôle avant export :

À COLLER

```
Passe les objets de la collection COL_Export_Kit à la liste de contrôle du pack
(validation-checklist) et liste ce qui ne passe pas : échelle, pivots,
transformations appliquées, nommage, nombre de triangles. N'exporte rien.
```

Si tu as activé le mode sûr et qu'un script est refusé, dis-le à Claude :

À COLLER

```
Le mode sûr de l'extension est activé : réécris ce script sans os ni open, avec
les fonctions de chemin de Blender.
```

---

11. Ce que ça ne fait pas encore

- Un monde complet. Claude l'a dit : des mois de production. En une session, tu obtiens un kit et une rue en blocs, à détailler ensuite.
- Des personnages finis. La skill character-artist décrit la méthode (proportions, topologie du visage), mais construire un corps en Python, bloc par bloc, reste lent. L'extension sait générer un modèle par IA, avec tes propres clés API (Meshy, Tripo, Hunyuan3D, Hyper3D Rodin) ou par son offre Premium, de 10 $ à 100 $ par mois au 11 novembre.
- Des textures peintes. Claude écrit des matériaux procéduraux et peut importer des textures de Poly Haven. Le reste passe par toi ou par un générateur payant.
- Une preuve que le pack améliore le résultat. Aucun avant/après publié au 11 novembre.
- Le travail dans Unreal. Tu importes les FBX, tu vérifies l'échelle (un mur de 4 m doit mesurer 400 cm, rappelle le brief) et tu assembles. Unreal reconnaît à l'import les collisions nommées UCX_ suivies du nom du mesh, d'après sa documentation.
- Un suivi actif. Un seul contributeur, des skills inchangées depuis le 10 juillet, le ticket du .mcp.json sans réponse. Si une version 1.4 sort, relis-la avant de mettre à jour.

---

12. Le cadre légal, là où ça compte

Textes relus le 11 novembre 2026. Ça t'aide à repérer les points d'attention, ça ne remplace pas un avocat.

Le pack et l'extension : licence MIT

Tu peux utiliser, copier et modifier les skills, y compris pour un usage commercial. Si tu les redistribues, garde la mention de copyright et le texte de la licence (« The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software »). Aucune garantie : « AS IS ». L'extension est sous la même licence ; ses conditions d'utilisation portent sur les données collectées (partie 6).

Blender : la GPL pour le logiciel, tes créations pour toi

Blender est sous licence GNU GPL, et sa page de licence tranche pour ce que tu produis : « What you create with Blender is your sole property. » Images, vidéos, fichiers .blend : usage commercial compris.

La GPL compte dans un cas : si tu publies un module complémentaire, un script Python qui se sert de l'API de Blender. Il doit alors être partagé sous une licence compatible GPL ; tu peux le vendre, l'acheteur le reçoit sous GPL. Les scripts que Claude écrit pour ton usage ne sont pas

concernés tant que tu ne les publies pas. Et pour ce qui sort de Claude, les conditions d'Anthropic pour les particuliers (en vigueur depuis le 8 octobre 2025) te cèdent « all our right, title, and interest (if any) in Outputs ».

La marque Blender

Blender est une marque déposée de la Blender Foundation dans l'Union européenne et aux États-Unis, et le nom comme le logo sont hors GPL. Un tiers peut utiliser le logo pour désigner le logiciel ou le créditer, en second plan derrière sa propre marque, jamais comme son propre logo. « Réalisé avec Blender » sur ton portfolio, oui. Le logo Blender en tête de ton offre commerciale, non.

Les ressources importées par l'extension

- Poly Haven : tout est en CC0. « You can use our assets for any purpose, including commercial work. You do not need to give credit or attribution when using them. »
- Poly Pizza : environ 69 % du catalogue est en CC-BY, qui oblige à créditer l'auteur partout où le modèle apparaît, selon le README de l'extension, qui enregistre la ligne de crédit dans ton fichier. Filtre en CC0 pour t'en passer.
- Sketchfab : chaque modèle a sa propre licence, à lire avant un projet commercial.

---

13. La méthode

Guide écrit le 11 novembre 2026. Le dépôt du pack a été cloné en lecture seule et recompté en Python. Statistiques, commits et tickets lus par l'API GitHub entre 15 h 13 et 15 h 25, heure de Paris. Pour l'extension : README, conditions d'utilisation, code du serveur, de la télémétrie et du mode sûr, script d'installation, tous lus sans être lancés. Documentation de Claude Code, blender.org, Poly Haven, catalogue officiel des plugins Claude lus le même après-midi ; page de prix de Claude rendue depuis la France à 14 h 50.

Rien n'a été installé ni lancé : ni Blender, ni uv, ni le pack, ni l'extension, ni le script de Claude, dont l'assemblage a été rejoué en Python pour la vidéo. Les commandes sont recopiées des README ; celles qu'on a adaptées sont signalées.

Ce guide décrit l'état du 11 novembre 2026. Si une commande ou un prix a changé le jour où tu lis, crois la page officielle.

Sources

- Le pack : arjun988/blender-skills, son ticket #4 sur le .mcp.json et son ticket #2 sur l'avant/après, relus le 11 novembre 2026.
- L'extension : ahujasid/mcp-for-blender, ses conditions d'utilisation, son offre Premium et sa démo du 11 mars 2025.
- Documentation de Claude Code : MCP, skills, référence des plugins.

- Les tarifs de Claude, relevés depuis la France le 11 novembre 2026, et les conditions d'Anthropic pour les particuliers.
- Blender : téléchargement, licence, logo et marque.
- La licence de Poly Haven.
- L'import FBX des static meshes dans Unreal Engine, pour le nommage des collisions.
- uv, l'installation.