

Pipeline complet pour la création d'images et vidéos storyboardées

Introduction et Vue d'Ensemble

Ce pipeline professionnel vise à transformer rapidement des storyboards en images haute qualité (puis en vidéos animatiques), en alliant **Windsurf** pour l'orchestration, **ComfyUI** en local pour l'inférence visuelle, et l'API **DALL-E** en complément. L'objectif est de garantir une exécution rapide, une grande **qualité visuelle**, une **flexibilité de style** (ex. *style Déclics ombre chinoise*, *style labo réaliste*), ainsi qu'une **fiabilité** dans la production. Chaque étape du pipeline est structurée pour préserver la cohérence du storyboard d'origine tout en appliquant le style souhaité de manière contrôlée.

Orchestration avec Madsea et Indexation des Storyboards

Madsea joue le rôle de chef d'orchestre du pipeline. Il permet :

- **Upload et découpage des storyboards** : Importez les planches de storyboard (par exemple un PDF ou des images séquentielles) dans Madsea qui va extraire chaque case/illustration individuellement. Ces images sont automatiquement **indexées** avec une nomenclature stricte, par exemple `E202_SQ0010-0010_AI-concept_v0001.jpg`.

exple E202 : on identifie la saison + épisode

SQ0010 : identifiant de la séquence

0010 : identifiant du plan

Layout : la tâche en question

v0001 : la version. v minuscule obligatoire et 4 chiffres

.jpeg : extension du fichier, vous l'avez deviné

- Cette convention de nommage contient l'ID d'épisode/projet (ici `E202`), le numéro de séquence et de plan (`SQ0010-0010` pour séquence 0010, plan 0010), le type d'image (`AI-concept`) et le numéro de version (`v0001`). Un nommage clair et cohérent facilite le suivi et la collaboration en production ([Animation Naming Conventions and Folder Structures for Game Development | by Nick Rodgers | Medium](#)), tout en permettant à Madsea de repérer automatiquement les éléments

(ex: *E202* pour l'épisode 202, *SQ0010-0010* pour le 1er plan de la séquence 001, etc.).

- **Gestion de version** : À chaque itération ou nouvelle génération d'image, Madsea incrémente la version dans le nom de fichier (passant à *v0002*, *v0003*, etc.), assurant un historique traçable sans écraser les précédents essais.
- **Sélection des images à traiter** : Via l'interface Madsea, l'utilisateur peut sélectionner **individuellement ou par lots** les images du storyboard à transformer en visuels finalisés. Par exemple, on peut choisir tous les plans d'une séquence donnée, ou seulement certains plans clés à retravailler. Cette souplesse permet de concentrer la puissance de calcul sur les images prioritaires. Une fois la sélection faite, Madsea préparera un **file de traitement** pour ComfyUI.

(À ce stade, toutes les images sources sont prêtes, correctement nommées et organisées. Madsea passe alors la main à l'étape d'inférence en communiquant la liste des images sélectionnées et les paramètres désirés.)

Calibration du Style sur ComfyUI (Style « *Déclics* », *Labo réaliste*, etc.)

Sur la machine locale dédiée à l'inférence, **ComfyUI** est utilisé pour générer les images dans le style artistique souhaité. ComfyUI est un outil nodal pour Stable Diffusion qui permet de construire des **workflows avancés** combinant modèles, ControlNets, LoRA, etc., de manière flexible. On y configure le style de rendu via plusieurs leviers :

- **Style Adapter (Adaptateur de style)** : Cette extension (issue des T2I-Adapters) permet d'injecter dans la pipeline une image de référence représentant le style visuel cible (ex: *une image d'exemple du style « Déclics » ombre chinoise*). Le Style Adapter va conditionner Stable Diffusion pour imiter les caractéristiques picturales de cette référence, sans altérer la composition de l'image d'entrée. C'est une approche *sans entraînement préalable* qui cherche à **dissocier le style du contenu**, en se branchant sur le modèle un peu comme un ControlNet dédié au style ([Replacing LoRA With a Generic Style Adapter in Stable Diffusion | Astrid Busser-Casas](#)). Bien que cette méthode ne rivalise pas encore totalement avec la qualité d'un LoRA entraîné spécifiquement ([Replacing LoRA With a Generic Style Adapter in Stable Diffusion | Astrid Busser-Casas](#)), elle offre une solution rapide pour approcher un style donné. Dans notre pipeline, on l'utilisera pour le style « *Déclics* » afin d'obtenir l'effet *ombre chinoise* (silhouettes nettes, contrastes forts, fonds unis) en se basant sur des visuels de référence de ce style.
- **LoRA de style** : Pour une fidélité de style maximale, on prévoit d'intégrer des **LoRA** (Low-Rank Adaptation) entraînés sur nos styles spécifiques. Un LoRA est un module additionnel contenant les deltas de poids du modèle principal, entraîné sur un petit jeu d'images représentatives d'un style ou d'un univers visuel. Par exemple, si le style « *Déclics* » nécessite des détails particuliers non atteints par le Style Adapter

seul, on peut entraîner un LoRA sur une dizaine d'images *Déclics* haute qualité. Ce LoRA appliqué sur le modèle Stable Diffusion permettra de reproduire ce style de façon cohérente. L'avantage est qu'un LoRA est léger et modulable : on peut l'activer/désactiver ou ajuster sa **pondération** (par ex. poids 0.8 pour un effet plus subtil ou 1.2 pour renforcer le style). Madsea orchestre l'activation du LoRA via ComfyUI (en chargeant le LoRA dans le pipeline pour les images concernées). À *noter*: un Style Adapter générique sera utilisé en amont pour guider le style, mais le LoRA, s'il est disponible, affinera le résultat avec une qualité potentiellement supérieure ([Replacing LoRA With a Generic Style Adapter in Stable Diffusion | Astrid Busser-Casas](#)).

- **Prompting stylisé** : En plus des méthodes ci-dessus, chaque image est générée avec un **prompt textuel** adapté au style voulu. Par exemple, pour *l'ambiance laboratoire réaliste*, le prompt comprendra des indications comme « *laboratoire scientifique réaliste, éclairage néon doux, ambiance sombre, style cinéma* ». Pour *le style ombre chinoise*, le prompt pourra mentionner « *silhouette en contre-jour, décor minimaliste, contraste élevé, style théâtre d'ombres* », etc. Ces indications textuelles, combinées aux adaptateurs de style ou LoRA, renforcent la direction artistique imposée à l'IA.

En calibrant ainsi le style dans ComfyUI, on garantit que chaque image générée reflète fidèlement la **direction artistique** définie (que ce soit un style très stylisé *ombre chinoise* ou du photoréalisme pur) avant de passer aux aspects photoréalistes plus techniques.

Maintien de la Composition avec ControlNet (Depth, Normal, Edge)

Pour **respecter le storyboard d'origine** – c'est-à-dire la composition, le cadrage, les silhouettes et la disposition des éléments – le pipeline fait un usage intensif de **ControlNet** dans ComfyUI. ControlNet permet d'imposer des contraintes visuelles au modèle de génération en lui fournissant des guides dérivés de l'image de storyboard :

- **ControlNet Depth (Profondeur)** : On extrait une **carte de profondeur** de chaque image de storyboard (via un préprocesseur MiDaS ou autre dans ComfyUI). Cette carte de profondeur représente la distance des objets par rapport à la caméra et donne une idée de la perspective de la scène. En fournissant ce *Depth Map* au modèle via un ControlNet dédié, on s'assure de conserver la disposition spatiale réaliste des éléments dans l'image générée (plans avant/arrière, échelle des objets) ([ControlNET Integration in ComfyUI | Medium](#)). Cela est crucial surtout pour les scènes complexes ou avec de la profondeur (ex: un personnage au premier plan et un décor en arrière-plan). Le Depth ControlNet **calibre le modèle sur la géométrie de la scène**, évitant par exemple qu'un objet censé être loin n'apparaisse trop grand ou flou incorrectement.
- **ControlNet Normal (Normales)** : À partir de la profondeur, on peut également déduire une **carte de normales** (orientations de surfaces) – souvent calculée par un

modèle annexe (par ex. *NormalBae* ou *Normal Midas* qui utilise MiDaS pour estimer les normales ([Mastering ComfyUI ControlNet: A Complete Guide - RunComfy](#))).

Fournir les normales en entrée via ControlNet aide le modèle à comprendre l'orientation des surfaces et la direction des volumes dans la scène. Par exemple, pour un décor de laboratoire en 3D, les surfaces des tables, des murs, etc., gardent ainsi leurs orientations cohérentes dans l'image générée. Le Normal map renforce le réalisme en conservant l'éclairage cohérent avec la géométrie (utile lorsqu'on ajoutera des effets de lumière comme le rim light).

- **ControlNet Edge/Line (Contours)** : Si le storyboard est dessiné avec des contours clairs (line art) ou que l'on dispose d'un crayonné, on utilise un ControlNet d'edges. Un préprocesseur Canny ou HED **détecte les bords et contours** du dessin de storyboard ([ControlNET Integration in ComfyUI | Medium](#)). En injectant ces *edges* dans Stable Diffusion via ControlNet, on oblige l'IA à respecter scrupuleusement le cadrage, les formes des sujets et leur pose. Par exemple, la silhouette d'un personnage en ombre chinoise sera conservée telle quelle, mais remplie de textures ou d'éclairage réaliste par le modèle. Pour un décor, les lignes principales (horizon, structures) resteront à la même place. Ce filet de sécurité garantit que malgré l'application d'un style nouveau, la **composition reste identique** à la vision du storyboard.

Chaque ControlNet est configuré avec une **influence (weight)** appropriée – typiquement une force élevée (aux alentours de 1.0) pour la structure de base (edges, depth) afin d'avoir un maximum de fidélité au layout ([ControlNET Integration in ComfyUI | Medium](#)). On peut combiner plusieurs ControlNets simultanément (par ex. Depth + Edge ensemble) car ComfyUI supporte le **multi-Conditioning** ([ControlNET Integration in ComfyUI | Medium](#)). Cela permet d'obtenir un résultat qui suit le plan du storyboard (grâce aux edges/depth) tout en adoptant le style et le réalisme désirés via le modèle et le style adapter. *En somme, les ControlNets agissent comme la « main du storyboarder » qui guide l'IA pour ne pas s'écarter de la scène prévue.*

Choix des Modèles d'IA (RealVisXL, SDXL, DreamShaper, etc.)

La qualité visuelle et le style dépendent fortement du **modèle de diffusion** utilisé. Notre pipeline prévoit d'employer différents modèles Stable Diffusion, sélectionnés selon le rendu recherché :

- **Stable Diffusion XL (SDXL)** : SDXL est un modèle de base de nouvelle génération, connu pour sa haute résolution native et sa meilleure fidélité aux détails par rapport aux modèles 1.5 classiques. Il sert de *backbone* polyvalent pour obtenir des images nettes et riches en détails. Nous l'utiliserons par exemple pour des scènes complexes où la résolution et la cohérence globale sont prioritaires. SDXL excelle dans le photoréalisme tout en restant flexible pour des styles variés si bien conditionné.

- **Realistic Vision (RealVisXL)** : *Realistic Vision* est un checkpoint finement entraîné pour le **photoréalisme**. Ce modèle (versions récentes v5, v6) produit des images pouvant rivaliser avec des photos, en particulier pour des sujets humains, décors réels et éclairages naturels. Son entraînement orienté « real » en fait un choix idéal pour le *style labo réaliste* par exemple. *RealVisXL* est en fait l'utilisation de Realistic Vision sur une base XL (d'où l'abréviation) pour combiner les atouts de SDXL et du réalisme accru. Comme l'indique sa documentation, ce modèle « *vise le réalisme et le photoréalisme* » ([SG161222/Realistic_Vision_V6.0_B1_noVAE · Hugging Face](https://github.com/SG161222/Realistic_Vision_V6.0_B1_noVAE)). Il est donc recommandé pour toutes les images où l'on souhaite un rendu *cinéma réel*. Avec Realistic Vision, on appliquera éventuellement une VAE améliorée pour maximiser la qualité (selon les conseils de son auteur) et on prêter attention aux résolutions optimales (ex : ~1024×768) pour éviter les artefacts.
- **DreamShaper** : DreamShaper est un modèle orienté *illustration réaliste stylisée*, souvent apprécié pour obtenir un rendu artistique tout en conservant une base réaliste. Il peut servir lorsque l'on souhaite un certain cachet visuel ou une patte graphique plus prononcée que le pur réalisme. Par exemple, pour générer des variations créatives d'un plan (*variantes de plans*), ou pour un style intermédiaire (mi-réaliste mi-artistique), DreamShaper apporte de la diversité. Il combine bien avec des LoRAs de style pour pousser encore son orientation visuelle.
- **Autres modèles et fine-tunes** : Le pipeline reste flexible – on peut intégrer d'autres checkpoints spécifiques si besoin. Par exemple, un modèle entraîné sur des cinématiques de film, ou un modèle spécialisé dans un type de décor (urbain, nature) pourrait être chargé dans ComfyUI pour quelques plans ciblés. Grâce à Madsea le changement de modèle par plan ou par lot est scripté : on définit pour chaque séquence quel modèle utiliser. Cette adaptabilité garantit qu'on utilise l'outil optimal pour chaque contexte (ex : Realistic Vision pour personnages réels, un modèle *toon shader* si on voulait un rendu BD, etc.).

Chaque modèle étant pré-chargé en local, le basculement est rapide. ComfyUI permet de **charger dynamiquement** le checkpoint voulu pour chaque lot d'images, assurant une transition fluide dans la chaîne de production.

Réglages précis pour un rendu cinéma photoréaliste

Afin d'obtenir un rendu final digne d'une production audiovisuelle (*effet cinéma*), nous appliquons des réglages fins dans ComfyUI :

- **Samplers avancés** : On utilise des algorithmes de sampling performants tels que *DPM++ 2M Karras* ou *DPM++ SDE* pour la génération d'images. Ces samplers offrent une bonne exploration de l'espace latent et une convergence stable vers des résultats détaillés. Par expérience, *DPM++ 2M Karras* à ~30 pas (steps) peut suffire pour beaucoup de cas, mais pour maximiser le photoréalisme on peut monter à 50 pas ou plus ([SG161222/Realistic_Vision_V6.0_B1_noVAE · Hugging Face](https://github.com/SG161222/Realistic_Vision_V6.0_B1_noVAE)). Cela permet de capturer des nuances fines (textures de peau, reflets, etc.) au prix d'un

temps de calcul un peu plus élevé – acceptable dans un contexte batch automatisé. Pour les styles plus simples (ombre chinoise), ~20-30 pas peuvent suffire étant donné la relative simplicité de l'image (grands aplats).

- **Guidance Scale (CFG)** : Le **coefficient de guidance** est ajusté pour trouver un équilibre entre fidélité au prompt et liberté de variation. En général, une valeur moyenne (aux alentours de 7 à 9) est utilisée pour du photoréalisme, ce qui force l'IA à respecter la description (et le style imposé) sans tomber dans l'excès (ce qui pourrait créer des artefacts ou un aspect trop figé). Si l'on travaille en mode *img2img* avec l'image de storyboard en référence, on tendra à réduire légèrement la guidance (et à utiliser un **strength** modéré sur l'image, par ex. *denoise strength* = 0.5-0.6) pour laisser le modèle recréer les détails tout en gardant la structure globale.
- **Résolution et upscaling** : On génère idéalement les images à une résolution suffisante pour la prévisualisation (ex : 1024×576 ou 1280×720 pour un storyboard 16/9). Ensuite, un *upscaling* peut être appliqué pour obtenir du **Full HD ou 4K** si requis par le montage final. ComfyUI intègre des nodes d'upscalers AI (type ESRGAN, **4x-UltraSharp** etc.) ou la technique du *Hires. fix* (recalcule l'image à plus haute résolution en fin de diffusion) ([SG161222/Realistic_Vision_V6.0_B1_noVAE · Hugging Face](#)) ([SG161222/Realistic_Vision_V6.0_B1_noVAE · Hugging Face](#)). Nous activerons Hires Fix pour les plans larges ou riches en détail afin d'améliorer la netteté sans introduire de différences de composition. Par exemple, une image initiale en 768×432 peut être upscalée 2× en fin de processus pour un rendu 1536×864 propre, en utilisant ~0.3 de denoise sur l'upscale pour ajouter du détail sans dénaturer l'image ([SG161222/Realistic_Vision_V6.0_B1_noVAE · Hugging Face](#)).
- **Éclairage cinématographique réaliste** : Le **prompt** sera enrichi de termes décrivant un éclairage de cinéma. On ajoutera systématiquement des indications comme « *éclairage tridimensionnel réaliste, rim light, global illumination, ombres portées douces, haute gamme dynamique* ». Le *rim light* (liseré de lumière en bordure des sujets) apporte du volume et fera ressortir les silhouettes, ce qui est parfait pour un style ombre chinoise modernisé. Le **bokeh** (flou d'arrière-plan) sera obtenu en mentionnant « *profondeur de champ réduite, bokeh cinématographique* », simulant un objectif ouvert (on peut même préciser *f/1.8* par ex.). Ces éléments, combinés à la carte de profondeur du ControlNet, produiront un flou réaliste en arrière-plan tout en gardant le sujet net.
- **Détails photoréalistes** : Pour le style labo réaliste, on veille à inclure dans le prompt des détails concrets : ex « *reflets sur les surfaces métalliques, légèrement poussiéreux, néons avec bloom léger, rendu photoréaliste* ». En négatif, on exclut « *effet cartoon, dessin, CGI, lowres* » pour éviter tout aspect non réaliste ([SG161222/Realistic_Vision_V6.0_B1_noVAE · Hugging Face](#)). Ces *negative prompts* sont importants pour pousser l'IA à éliminer les attributs de rendu indésirables (traits de dessin, défauts anatomiques, etc.).
- **Cohérence sur une séquence** : Si plusieurs images successives (plans d'une même scène) sont générées, on peut fixer certains paramètres pour la cohérence

visuelle : par exemple, réutiliser le même **seed de bruit** de départ pour garder une cohérence dans le grain ou la colorimétrie, ou bien appliquer en post-production un **color grading** identique sur la série d'images. Madsea peut faciliter cela en maintenant un seed constant pour une série d'images si on le souhaite, ou en appliquant un **script de post-traitement** (via ImageMagick ou autre) sur toutes les images exportées pour harmoniser les couleurs/exposition. L'important est qu'une fois importées dans la vidéo finale, les images ne sautent pas d'un style/lumière à l'autre de façon incohérente.

En ajustant ces paramètres, on obtient des images qui **ressemblent à des captures de film**, prêtes à être montées comme un vrai film storyboardé, avec le bon grain, la bonne lumière et une définition suffisante.

Entraînement / Fine-tuning de nouveaux styles (LoRA personnalisés)

Dans le cas où un style visuel spécifique serait requis et que ni les prompts ni les adaptateurs de style n'arrivent à s'en approcher assez, le pipeline prévoit une étape de **fine-tuning** dédiée. Plutôt que de ré-entraîner un modèle complet (coûteux en temps), on opte pour la création de **LoRA sur mesure** :

- **Collecte de données de style** : On rassemble une petite galerie d'images (idéalement 10 à 30) illustrant le style voulu. Par exemple, pour un nouveau projet, cela pourrait être des concept arts existants, des extraits d'œuvres visuelles de référence, etc. Ces images doivent partager des caractéristiques stylistiques (couleurs, tracés, textures) que l'on souhaite apprendre à l'IA.
- **Fine-tuning LoRA** : À l'aide d'outils d'entraînement (type *Kohya LoRA trainer* ou équivalent intégré), on entraîne un LoRA sur ces images. On utilise des *prompts* descriptifs génériques pour que le LoRA apprenne le style plutôt qu'un contenu spécifique. En quelques heures de calcul (sur GPU local ou cloud), on obtient un fichier LoRA (quelques dizaines de Mo).
- **Intégration** : Ce LoRA est ensuite chargé dans ComfyUI pour les générations. Madsea peut proposer une interface pour gérer la bibliothèque de LoRAs (ex: *LoRA StyleDéclics_v1*, *LoRA_NouveauStyle_v1*, etc.) et les associer à certaines images ou séquences. Lors de la génération, ComfyUI appliquera le LoRA en plus du modèle de base. Si besoin, plusieurs LoRAs peuvent être combinés (par ex. un LoRA de style + un LoRA de personnage pour garder un character design constant). On veillera toutefois à modérer les poids pour éviter les conflits.

En procédant ainsi, le pipeline reste **évolutif** : de nouveaux styles peuvent être incorporés à la volée sans tout reconfigurer. La *training pipeline* pourrait même être orchestrée via Madsea (ex: un module **"train style"** qui, une fois donné un dossier d'images de référence,

lance un entraînement LoRA et l'enregistre). Cela garantit une **flexibilité** maximale : quel que soit le défi artistique, on peut outiller l'IA pour y répondre.

Usage de l'API DALL-E en cloud en complément

Bien que le cœur du pipeline repose sur Stable Diffusion en local (via ComfyUI), nous prévoyons un **filet de sécurité créatif** avec l'**API DALL-E** d'OpenAI, utilisable à tout moment en parallèle :

- **Scénarios d'utilisation** : DALL-E 3 peut être sollicité pour générer des idées visuelles additionnelles ou débloquer des cas difficiles. Par exemple, si malgré nos LoRAs et prompts un plan précis n'atteint pas le rendu désiré (compositing complexe, élément inhabituel), on peut envoyer la requête à DALL-E pour voir une alternative. DALL-E excelle souvent dans la créativité et la compréhension fine des instructions, ce qui peut donner un résultat exploitable directement ou comme **base** à affiner.
- **Intégration orchestrée** : Madsea intègre un appel API qui envoie le prompt (et éventuellement l'image de storyboard en input si DALL-E supporte l'*inpainting* ou *outpainting* contextuel) au cloud. On peut définir par exemple un **flag** dans Madsea (comme `use_dalle=True` sur une opération) pour détourner la requête vers DALL-E plutôt que ComfyUI. L'image retournée par DALL-E sera récupérée par Madsea, nommée selon notre convention, et placée dans le lot de résultats. Ainsi, pour l'utilisateur, la différence est transparente, si ce n'est possiblement un léger délai dû à l'appel réseau et les coûts associés.
- **Complémentarité** : Les images DALL-E obtenues peuvent soit être utilisées telles quelles si elles conviennent, soit servir de **matériel de départ** pour Stable Diffusion. En effet, on pourrait ré-injecter une image DALL-E dans ComfyUI en *img2img* ou via ControlNet (par ex. extraire sa profondeur) afin de garder l'uniformité de style avec le reste du pipeline. Par exemple, DALL-E génère une composition très juste d'une scène futuriste difficile à obtenir, on peut ensuite la passer dans SDXL + LoRA style *Déclics* pour l'harmoniser avec les autres images.

En résumé, l'API DALL-E agit comme une **solution de secours** et d'inspiration. Elle ajoute de la fiabilité au pipeline en garantissant qu'on a toujours une image exploitable, même en cas de difficulté, sans interrompre le flux de travail.

Traitement des Images en Lots vs Individuellement

Un des atouts de ce pipeline est sa capacité à traiter un grand nombre d'images de façon automatisée tout en permettant des ajustements au cas par cas :

- **Traitement par lots accéléré** : Madsea permet de lancer le pipeline ComfyUI sur un **ensemble d'images sélectionnées** en une seule commande. Par exemple, on peut générer *tous les plans de la séquence 0010* dans le style labo réaliste en un clic.

ComfyUI exécutera alors la génération en série (ou en parallèle si plusieurs GPU) pour chaque image, en appliquant le même modèle, les mêmes ControlNets et prompts. Cela assure une **cohérence de style** sur le lot et un gain de temps énorme (pas besoin d'intervenir entre chaque image). Les logs de chaque image (seed, prompt, paramètres) sont enregistrés pour suivi. Ce mode batch est privilégié quand on est confiants dans les réglages et que le style est uniforme sur tout le lot.

- **Affinage individuel** : En parallèle, chaque image peut être traitée **individuellement** si nécessaire. Supposons qu'un plan clé requiert plus d'attention (par ex. un gros plan sur un personnage principal où l'expression doit être parfaite) – l'utilisateur peut choisir de générer plusieurs variantes de cette image. Madsea permet par exemple de spécifier **variants=5** pour une image, ce qui lancera 5 générations avec des seeds différents ou de légères variations de prompt. L'utilisateur pourra ensuite choisir la meilleure. De plus, via l'interface ComfyUI (accès manuel ou via preview Madsea), on peut ajuster **interactivement** les nodes pour une image donnée (par ex. renforcer un ControlNet, changer un mot de prompt) et relancer uniquement celle-ci. Ce travail fin unitaire peut se faire sans impacter le reste du batch.
- **Gestion des retours** : Après génération, Madsea présente les résultats (chaque image marquée de son nom/numéro). L'utilisateur peut alors marquer celles qui sont validées, et celles à relancer éventuellement avec d'autres réglages. Le pipeline encourage un cycle itératif : on peut relancer un second batch uniquement sur les images non satisfaisantes en ajustant les paramètres (par ex. augmenter la guidance pour plus de précision, ou appeler DALL-E en alternative comme mentionné). Ce système **itératif** assure que l'on converge vers un rendu satisfaisant pour *chaque* image, tout en tirant parti de l'automatisation sur l'ensemble.

En somme, la combinaison batch + individuel apporte **rapidité** (génération massive automatique) et **contrôle de la qualité** (ajustements ciblés), ce qui est crucial dans un contexte de production où le temps est compté mais où chaque image compte.

Export final et Compatibilité avec Krea/Kling/Runway

Une fois les images finalisées, le pipeline s'occupe de l'**exportation** dans les formats et emplacements requis pour la suite du workflow audiovisuel, notamment pour un usage dans *Krea*, *Kling* ou *Runway* (outils/platformes de post-production et de montage collaboratif) :

- **Format d'image standard** : Toutes les images générées sont exportées en **PNG haute qualité** (format sans perte idéal pour conserver les détails et faciliter d'éventuels retouches ultérieures). Si l'espace de stockage est une contrainte, on peut basculer en JPEG qualité maximale, mais en production on privilégie le PNG 16 bits pour garder la latitude en colorimétrie. La résolution cible peut être Full HD (1920×1080) vu que ces images sont destinées à un animatique vidéo, voire 4K si on anticipe des zooms/coupes au montage. Madsea peut effectuer une conversion batch si nécessaire (toutes les images d'un lot passées de .jpg provisoire à .png final

par ex.).

- **Nomenclature et métadonnées** : L'export conserve la nomenclature stricte (E202_SQ0010-0010_AI-concept_v0002.png, etc.) pour que chaque fichier s'intègre sans ambiguïté dans la bibliothèque du projet. De plus, Madsea peut attacher des **métadonnées** (EXIF/IPTC) mentionnant le modèle IA, le seed, le prompt utilisé – ces infos pouvant être utiles plus tard pour régénérer ou ajuster des images précises. Ces métadonnées n'interfèrent pas avec Krea/Kling, mais offrent de la traçabilité en interne.

En conclusion, l'export final transforme le travail effectué en amont en livrables concrets pour la suite du projet. Chaque image est prête à être utilisée, ce qui clôture le pipeline d'une manière fluide du storyboard initial jusqu'à l'animation vidéo finale.

Exemples de *prompts* Madsea pour lancer les opérations

Voici quelques exemples de commandes (ou instructions) que l'on pourrait utiliser dans Madsea pour piloter ce pipeline de bout en bout :

Import du storyboard et indexation :

```
Madsea> ImporterStoryboard "E202_storyboards.pdf" --decouper
```

- *Action* : Madsea importe le PDF du storyboard de l'épisode 202 et en extrait chaque case en image individuelle. Les fichiers sont nommés automatiquement selon la convention (par ex. `E202_SQ0001-0005_Concept_v0001.jpg`, etc.) et rangés par séquence.

Sélection des plans à générer :

```
Madsea> SelectionnerPlans E202 SQ0010
```

- *Action* : Sélectionne tous les plans de la séquence 0010 de l'épisode 202 pour traitement. (On pourrait affiner en listant explicitement des plans, ex: `SelectionnerPlans E202 SQ0010-0010, SQ0010-0020` pour ne choisir que deux plans précis).

Configuration du style et du modèle :

Madsea> ConfigurerGeneration modele="RealVisXL", style="ombre chinoise", controlnets=["Depth","Edge"], sampler="DPM2M_Karras", steps=40, CFG=8

- *Action* : Définit les paramètres de génération à utiliser sur la prochaine opération. Ici on choisit le modèle **Realistic Vision XL**, on active le style « Ombre chinoise » (ce qui chargera le LoRA de style Déclics et/ou le Style Adapter correspondant), on active deux ControlNets (Profondeur et Edges) pour guider la composition, et on fixe le sampler et quelques réglages (40 étapes, guidance 8) optimisés pour du photoréalisme cinéma. Ces réglages resteront en vigueur tant qu'on ne les modifie pas ou jusqu'à la fin du batch.

Génération d'images (batch) :

Madsea> GenererImages selection

- *Action* : Lance ComfyUI pour générer les images de tous les plans sélectionnés, en appliquant les paramètres configurés. Chaque image de storyboard sélectionnée sera traitée : ComfyUI chargera l'image, appliquera le ControlNet de profondeur/edges, le modèle RealVisXL avec le style Ombre chinoise, et produira une nouvelle version enregistrée automatiquement (ex: **E202_SQ0010-0010_Concept_v0002.png**). Madsea affiche l'avancement plan par plan.

Générer des variantes pour un plan :

Madsea> GenererVariante "E202_SQ0010-0010_Concept_v0002.png" --nombre=3

- *Action* : Pour l'image spécifiée (plan 0010-0010 version2), lance la génération de 3 variantes supplémentaires (par ex. en changeant le *seed* aléatoire à chaque fois). Cela permet de comparer plusieurs propositions visuelles pour ce plan. Les fichiers sortiront comme **...v0003.png**, **...v0004.png**, etc. Madsea peut ensuite afficher ces 3 variations côte à côte pour choix.

Utilisation de DALL-E en fallback :

Madsea> GenererAvecDALLE "E202_SQ0010-0020_Concept_v0001.jpg"
--prompt_alt="laboratoire ultraréaliste style Labo déclics"

- *Action* : En cas de difficulté avec Stable Diffusion, cette commande envoie l'image de storyboard du plan 0020-0020 vers l'API DALL-E avec un prompt alternatif. L'image retournée par DALL-E sera sauvegardée comme **..._v0002_DALLE.png** et pourra être ensuite intégrée au lot. Madsea gère l'appel API et la récupération du

fichier automatiquement.

Export des résultats :

Madsea> ExporterImages E202 --vers="serveur ftp"

- *Action* : Rassemble toutes les images générées pour l'épisode 202 et les copie dans le dossier (ou projet) connecté à serveur ftp. Cela assure que l'équipe post prod retrouve toutes les images à jour dans le serveur. Éventuellement, Madsea pourrait ici effectuer des conversions (vers PNG) ou un redimensionnement final si requis.

(Les syntaxes ci-dessus sont indicatives – en pratique l'interface Madsea pourrait être graphique ou en langage naturel. L'important est qu'on puisse enchaîner aisément ces commandes pour piloter chaque étape du pipeline.)

En synthèse, ce pipeline complet permet de passer du **storyboard 2D** à des **images finalisées** de qualité cinématographique de façon fluide. Madsea assure la **coordination** et la gestion de projet (organisation des fichiers, lancement des lots, suivi des versions), tandis que ComfyUI et ses modèles IA garantissent la **qualité visuelle** en combinant style et réalisme (grâce aux LoRA, Style Adapter et ControlNet qui préservent le contenu ([ControlNET Integration in ComfyUI | Medium](#)) ([ControlNET Integration in ComfyUI | Medium](#))). L'ajout de l'API DALL-E apporte une **assurance supplémentaire** pour résoudre les impasses créatives. Pensé pour un contexte de production audiovisuelle réel, chaque choix dans ce pipeline vise la **rapidité** (automatisation, batch processing), la **flexibilité** artistique (multiples styles, modèles interchangeables) et la **fiabilité** des résultats (contrôle de structure par ControlNet, naming clair, versions sauvegardées). Ce flux de travail donne à l'équipe la possibilité de générer un grand volume d'images cohérentes et époustouflantes en un temps record, tout en gardant la main sur la direction artistique jusqu'au bout. ([Animation Naming Conventions and Folder Structures for Game Development | by Nick Rodgers | Medium](#)) ([ControlNET Integration in ComfyUI | Medium](#))