

EpiGimp2.0



Cahier des charges & plan de livraison

Rédigé par : Apolline Fontaine et Léandre Godet

Pour : Epitech (client)

v1.5 : 10/03/2026

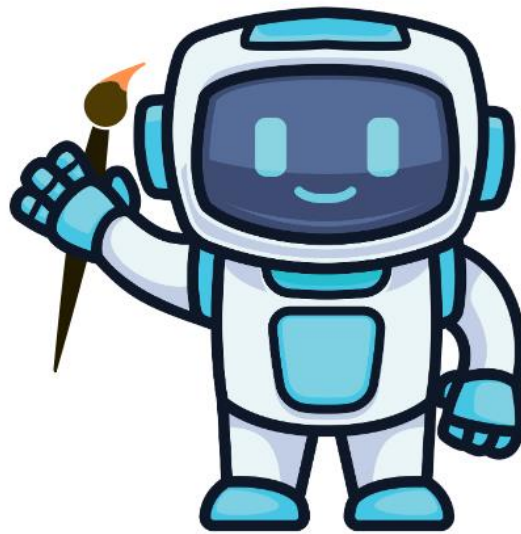
Sommaire

Historique des versions

Version	Date	Auteur	Changements principaux
1.0	20/10/2025	Apolline.F & Léandre.G	Création du document
1.1	22/10/2025	Apolline.F	Ajout des sections Architecture & .epg
1.2	03/11/2025	Apolline.F & Léandre.G	Ajout du lien vers la maquette Ajout de la section Hors-Périmètre Mise à jour : Correction des erreurs de mise en page Ajout des critères d'acceptations pour verrouiller et fusionner – cf. fonctionnalité 2
1.3	05/11/2025	Apolline.F	Mise à jour de la section architecture : Ajout du modèle et du pourquoi ce choix d'architecture Ajout d'illustration Mise à jour de la section rendu
1.3.2	12/11/2025	Apolline.F	Ajout du lien vers l'UML
1.4	23/12/2025	Apolline.F	Amélioration à la suite de la review : <ul style="list-style-type: none">- Détailler les fonctionnalités avec l'estimation / fonctionnalités- User Story
1.5	10/02/2026	Léandre.G	Mise à jour des dates et version et outils

Introduction

Le projet **EpiGimp2.0** vise à concevoir un éditeur d'images libre et multiplateforme offrant un socle réaliste de fonctionnalités inspirées de GIMP. Le développement se déroule sur 10 semaines, à raison de 3 jours par personne et par semaine.



Contexte

L'objectif est de livrer un **MVP solide** permettant les **modifications classiques d'images** (PNG/JPEG), la **gestion de calques**, avec une expérience fluide et des preuves de qualité (tests, CI, exécutable).

Parties prenantes

Public : étudiants/développeurs/designers souhaitant un outil simple.

OS cibles : Windows / Linux

Outils

Repo : GitHub

CI : GitHub Actions

Communication : Discord

Code qualité : Lint, Github Dependabot, Codecov

Technologies

C++20, CMake, QT 6.x

Performance & Benchmark

UI

Axe	Qt	GTK	Electron
Portabilité Win/Linux	★★★★★	★★☆☆☆	★★★★☆
Poids/Perf rendu 2D natif	★★★★★	★★★★☆	★★☆☆☆
Outils/Docs	★★★★★	★★★★☆	★★★★☆
Sources	(doc.qt.io)	(The GTK Team)	(electronjs.org)

Core

Critère	C++	Python	Rust
---------	-----	--------	------

Perf CPU	★★★★★ langage natif optimisé, base de référence des benches. (benchmarksgame-team.pages.debian.net)	★★☆☆☆ nettement plus lent en pur Python sur compute. (benchmarksgame-team.pages.debian.net)	★★★★★ proche C/C++ selon implémentation. (benchmarksgame-team.pages.debian.net)
Compat Qt (UI & paint)	★★★★★ Native : API Qt en C++ (QImage/QPainter, signaux/slots) sans overhead inter-langage. (doc.qt.io)	★★★☆☆ Bindings officiels : Qt for Python (PySide6). Bon pour l'UI, mais le compute lourd doit passer par des extensions C/C++ (ex: pybind11). (doc.qt.io)	★★☆☆☆ Interop via ponts : CXX-Qt (KDAB) permet Rust↔Qt, viable mais écosystème plus jeune ; souvent utilisé via FFI C depuis une app Qt C++. (kdab.github.io)
Interop Compute		pybind11 : expose C++ à Python (header-only). (Pybind11 Documentation)	CXX-Qt / FFI C : Rust ⇌ C++/Qt. (kdab.github.io)

UI : Qt 6 (Widgets + QPainter) pipeline de peinture natif, latence basse, portabilité Windows/Linux, empreinte plus faible qu'une UI web embarquée. Réfs : Qt Paint System / QPainter ; Qt Widgets. doc.qt.io+2doc.qt.io+2

Core : C++ perfs CPU de référence, intégration **directe** avec Qt (pas de coût de bindings), chemin simple pour I/O (PNG/JPEG) et historique/undo.

Périmètre

Must have

FICHIERS :

- ✓ Nouveau : Créer un document vide
- ✓ Ouvrir PNG/JPEG + .epg
- ✓ Enregistrer : .epg (format natif remplaçant .xcf)
- ✓ Export : PNG/JPEG.

CALQUES :

- ✓ Créer / Supprimer / Dupliquer
- ✓ Réordonner
- ✓ Fusionner
- ✓ Visibilité / Opacité
- ✓ Verrouiller (déplacement/ édition)

OUTILS :

- ✓ Pinceau : couleur, taille, forme, dureté, opacité
- ✓ Gomme : taille, forme, dureté, opacité
- ✓ Sélection rectangulaire : créer, déplacé et supprimer
- ✓ Pot de peinture

HISTORIQUE :

- ✓ Annuler / rétablir

Should have

OUTILS :

- ✓ Pipette
- ✓ Texte
- ✓ Grille
- ✓ Raccourcis clavier

Could have

OUTILS :

- ✓ Sélection libre (lasso)

FICHIERS :

- ✓ Formats supplémentaires export WebP.
- ✓ Ouvert récemment : Liste des 5 derniers fichiers

Hors périmètre

- ✗ Filtres
- ✗ Mode de fusion avancés pour les calques

Exigences fonctionnelles

F-01- Gestion des fichiers

Type : *Must have*

User story

EN TANT QU'UTILISATEUR JE VEUX CREER UN PROJET, OUVRIR DES IMAGES ET SAUVEGARDER MON TRAVAIL, AFIN DE REPRENDRE EXACTEMENT LA OU JE ME SUIS ARRETE ET EXPORTER EN PNG/JPEG.

Scope fonctionnel

- Nouveau projet (dimensions, fond) → calque initial vide
- Ouvrir PNG/JPEG → crée un calque à l'échelle ajustée (fit)
- Enregistrer **.epg** (ré-ouverture à l'identique : calques/ordre/visibilité/opacités)
- Export PNG/JPEG (rendu **aplati**)
- Gestion d'erreurs (fichier corrompu/format non géré) → message clair, pas de crash

Critères d'acceptation

1. **Nouveau projet** : l'utilisateur choisit largeur/hauteur/fond → le canvas s'affiche avec un calque initial vide.
2. **Ouvrir** : ouvrir un PNG/JPEG crée un **calque** à partir de l'image importée et l'affiche à l'échelle ajustée.
3. **Enregistrer (.epg)** : le fichier projet rouvre **à l'identique** (ordre des calques, visibilité, opacités).
4. **Exporter** (PNG/JPEG) : le rendu exporté **correspond au rendu écran aplati** (dimensions et canaux conformes).
5. **Erreurs** : un fichier corrompu/format non pris en charge n'entraîne **aucun crash** ; un message d'erreur explicite est affiché.

Estimation Total : ~4 j

- Conception format .epg (spec + loader/saver) : **1.5 j**
- Import PNG/JPEG + création calque + fit : **1 j**
- Export PNG/JPEG (qualité, aplati) : **0.5 j**
- Erreurs/boîtes de dialogue + tests : **1 j**

F-02- Calques

Type : *Must have*

User story

EN TANT QUE DESIGNER JE VEUX GERER DES CALQUES (CREER, DUPLIQUER, REORDONNER, FUSIONNER, VERROUILLER, REGLER VISIBILITE/OPACITE) AFIN DE COMPOSER MES ELEMENTS PROPREMENT ET DE FAÇON NON DESTRUCTIVE.

Scope fonctionnel

- Créer/Supprimer/Dupliquer
- Réordonner (drag & drop / Up/Down)
- Visibilité (œil), Opacité (0–100 %)
- Verrouiller (protège de toute modif ; visibilité/opacité autorisées)
- Fusionner (vers le bas, ou multi-sélection → un calque)
- Persistance dans **.epg**

Critères d'acceptation

1. Créer / Supprimer / Dupliquer / Verrouiller

- ✓ Les opérations apparaissent immédiatement dans la pile des calques **et** dans le rendu.
- ✓ L'historique enregistre une entrée par action (annuler/rétablir OK).
- ✓ Le nom, l'ordre, la visibilité, l'opacité et l'état **verrouillé** sont conservés à l'enregistrement **.epg** et à la réouverture.

2. Réordonner :

- ✓ Glisser-déposer/Utilisation des boutons 'Up' 'Down' d'un calque modifie le rendu immédiatement.
- ✓ L'ordre affiché dans la pile correspond à l'ordre de composition du rendu.

3. Visibilité :

- ✓ Cliquer sur l'icône « œil » masque/affiche le calque instantanément.

- ✓ La visibilité n’affecte pas les données du calque, uniquement le rendu.

4. **Opacité :**

- ✓ Un slider 0–100 % modifie le rendu pixel-à-pixel en temps réel.

5. **Fusionner :**

- ✓ Un seul calque sélectionné : la commande produit un seul calque résultant de la composition du calque sélectionné et de celui immédiatement inférieur, à l’identique du rendu visible pour ces deux calques.
- ✓ Plusieurs calques sélectionnés : produit un seul calque résultant de la sélection dont les pixels sont **exactement** ceux du rendu de la sélection.
- ✓ **Nom** : le calque résultant reçoit par défaut le nom “Fusion de N calques”.
- ✓ **Verrous** : la fusion **échoue** si au moins un calque impliqué est verrouillé et affiche un message clair.
- ✓ **Aucune** perte d’alpha/transparence ; le résultat superposé au reste du document est **visuellement identique** avant/après l’opération.

6. **Verrouiller :**

- ✓ Le verrou empêche toutes modifications (outils, déplacements, renommé, dupliqué, supprimé, fusion)
- ✓ Le verrou n’empêche pas la capacité à voir ou non un calque avec les fonctionnalités ‘Visibilité’ et ‘Opacité’.
- ✓ **UI** : l’icône de cadenas reflète l’état
- ✓ L’état verrouillé est sauvegardé dans le .epg et restauré à l’identique à l’ouverture

Estimation Total : ~5 j

- Pile de calques (modèle + UI) : **1.5 j**
- Réordonner + visibilité/opacité : **1 j**
- Verrouiller (règles + UI + persistance) : **1 j**

- Fusionner (2 calques & multi-sélection) : **1.5 j**

F-03- Outils d'édition

Type : *Must have*

1 – Pinceau

User story

EN TANT QU'ILLUSTRATEUR JE VEUX PEINDRE AVEC UN PINCEAU (TAILLE, DURETE, COULEUR) AFIN DE RETOUCHER RAPIDEMENT.

Critères d'acceptation

1. Dessiner modifie uniquement la zone sous le trait ;
2. Annuler restaure l'état précédent.

2 – Gomme

User story

EN TANT QU'ILLUSTRATEUR JE VEUX GOMMER AFIN DE RENDRE DES ZONES TRANSPARENTES. JE VEUX POUVOIR CHANGER LA DURETE DE LA GOMME AFIN DE RENDRE UNE COULEUR PARTIELLEMENT OU TOTALEMENT TRANSPARENTE. JE VEUX POUVOIR CHANGER LA TAILLE DE MA GOMME AFIN D'ADAPTER LA ZONE SUR LAQUELLE JE VEUX AGIR.

Critères d'acceptation

1. Rend transparent les pixels gommés (alpha 0)
2. Changer la taille de la gomme
3. Changer la dureté (alpha 0 à 100%)
4. Annuler restaure.

3 – Sélection rectangulaire

User story

EN TANT QU'ILLUSTRATEUR JE VEUX SELECTIONNER UNE ZONE RECTANGULAIRE AFIN DE LIMITER L'IMPACT DES OUTILS.

Critères d'acceptation

1. La zone sélectionnée limite l'action des outils à celle-ci.
2. "Add/Subtract" modifie correctement le masque de sélection.

4 – Pot de peinture

User story

EN TANT QU'ILLUSTRATEUR JE VEUX REMPLIR UNE ZONE AVEC TOLERANCE AFIN DE COLORER RAPIDEMENT DES ZONES POSSEDANT LE MEME RGB.

Critères d'acceptation

1. Le pot affecte la zone attendue selon le seuil,
2. En présence d'une sélection, **seule** la sélection est remplie.

Estimation Total : ~7 j

- Pinceau (performant, peinture tuile/dirty-rect) : **2.5 j**
- Gomme (alpha) : **1 j**
- Sélection rectangulaire (mask + UI) : **1.5 j**
- Pot de peinture (tolérance + limites sélection) : **2 j**

F-04- Historique (Annuler/Rétablir multi-niveaux)

Type : *Must have*

User story

EN TANT QU'UTILISATEUR JE VEUX POUVOIR ANNULER/RETABLIR MES ACTIONS AFIN D'EXPERIMENTER SANS RISQUE.

Critères d'acceptation

1. **Profondeur** : au moins **20** opérations successives peuvent être annulées/rétablies.
2. **Granularité** : chaque trait de pinceau/remplissage est une unité d'historique cohérente.

3. **Stabilité** : Anuler/Rétablir successifs ne dégradent pas la pile des calques (tests répétés OK).

Estimation : ~2 j

F-05- Pipette

Type : *Should have*

User story

EN TANT QU'UTILISATEUR JE VEUX PRELEVER LA COULEUR SOUS LE CURSEUR (COMPOSITE) AFIN DE REUTILISER EXACTEMENT LA TEINTE VUE.

Critères d'acceptation

1. **Prélèvement** : la couleur prélevée correspond au **rendu composite** sous le curseur.
2. **Affichage** : la couleur active de l'outil passe immédiatement à celle prélevée (aperçu dans l'UI).

Estimation : ~0.5 j

F-06-Texte

Type : *Should have*

User story

EN TANT QUE DESIGNER JE VEUX AJOUTER DU TEXTE (POLICE, TAILLE, COULEUR) SUR UN NOUVEAU CALQUE AFIN D'ANNOTER MES IMAGES.

Critères d'acceptation

1. **Création** : saisir du texte crée un calque indépendant avec rendu fidèle (police/antialias).
2. **Édition** : re-éditer le texte **tant que** le calque n'a pas été rasterisé définitivement (ou rasterisation à la création si simplification MVP).
3. **Export** : le texte apparaît correctement dans l'image exportée.

Estimation : ~2 j

F-07- Navigation & repères

Type : *Should have*

User story

EN TANT QU'UTILISATEUR JE VEUX ZOOMER ET DEPLACER LA VUE, ET VOIR LA TRANSPARENCE VIA UN DAMIER AFIN DE TRAVAILLER PRECISEMENT.

Critères d'acceptation

1. **Zoom** : niveaux $\times 0,25$, $\times 0.5$, $\times 1$, $\times 2$, $\times 4$, $\times 8$; le curseur reste centré sur la zone visée.
2. **Pan** : déplacement fluide du canvas (outil Main ou barre d'espace).
3. **Checkerboard** : motif visible uniquement dans les zones transparentes.

Estimation : ~1.5 j

F-08- Raccourcis clavier

Type : *Should have*

User story

EN TANT QU'UTILISATEUR JE VEUX DECLENCHER RAPIDEMENT LES ACTIONS PAR RACCOURCIS AFIN D'ACCELERER MON FLUX.

- Ctrl + Z/Y (annuler/rétablir)
- Ctrl + +/- (pour zoomer)
- R (Sélection rectangulaire)
- F (Sélection lasso)
- T (texte)
- Ctrl + N (Nouveau)
- Ctrl + O (Ouvrir)
- Ctrl + S (Enregistrer)
- P (pinceau)
- Maj + E (gomme)
- Maj + B (pot)
- A (aérographe)
- O (pipette)
- Ctrl + Maj + E (Exporter)
- Ctrl + W (Fermer)
- Ctrl + C/V (Copier)

Critères d'acceptation

1. **Mapping** : chaque raccourci déclenche l'action attendue et met à jour l'UI.

2. **Conflits** : aucun conflit avec les raccourcis système Qt par défaut pour ces actions.

Estimation : ~0.5 j

F-09- Sélection libre (lasso)

Type : *Could have*

User story

EN TANT QU'UTILISATEUR JE VEUX TRACER UNE SELECTION LASSO AFIN D'ISOLER DES ZONES IRREGULIERES.

Critères d'acceptation

1. **Précision** : les outils n'agissent que dans la zone lasso.
2. **Combinaison** : compatible avec Add/Subtract de la sélection rectangulaire.

Estimation : ~2 j

F-10- Formats supplémentaires (export WebP)

Type : *Could have*

User story

EN TANT QU'UTILISATEUR JE VEUX EXPORTER EN WEBP AFIN DE REDUIRE LA TAILLE DES IMAGES POUR LE WEB.

Critères d'acceptation

1. **Export** : le fichier WebP est lisible par un viewer tiers ; dimensions identiques au canvas.
2. **Qualité** : un slider de qualité produit des tailles de fichiers différentes ; le rendu varie conformément aux paramètres.

Estimation : 1 j

Exigences non fonctionnelles

NF-01- Performance

- **Ouverture** : PNG 1080p ouvert en **< 200 ms**.
- **Pinceau** : latence moyenne **< 50 ms** (mesurée sur 10 traits de 500 px).
- **Fréquence de rendu** : **≥ 30 FPS** en édition courante (1080p, 1–3 calques).

Estimation : ~1 j

NF-02-Robustesse & erreurs

- **Aucun crash** sur fichiers invalides (PNG/JPEG corrompus, .epg incomplet) ; messages d'erreur clairs.
- **Annuler/Rétablir** : **≥ 20** étapes sans fuite mémoire (vérifié avec sanitizers en build Debug).
- **Autosave** (*si activé plus tard*) : ne corrompt jamais le projet.

Estimation : en continue

NF-03- Portabilité

- **OS** : exécutable **Windows 10+** et **Linux x64**.
- **Packaging** : artefacts **ApplImage** (Linux) et **exe** (Windows) générés en CI.

Estimation : ~1.5j

NF-04-Qualité & CI/CD

- **CI vert** (build + tests) sur Windows & Linux.
- **Couverture** : **≥ 60 %** sur le noyau image avec gtest, **≥ 60 %** sur les composant QT avec QT Test.
- **Analyse statique** : clang-tidy sans erreurs bloquantes ; **style** via clang-format.

Estimation : ~2 j

NF-05-UX & accessibilité de base

- **Raccourcis** documentés, états d'outils visibles, curseur adapté (pinceau/gomme).
- **Démarrage** : application prête en **< 1 s** sur la machine de démo.
- **Thème** : mode sombre par défaut, UI lisible en 1080p.

Estimation : ~2 j

NF-06-Sécurité & données

- **.epg** valide et **tolérant** : ignore les clés inconnues, vérifie dimensions/canaux, refuse les contenus exécutables.
- **Aucune exécution** de code tiers (plugins).

Définition de terminé

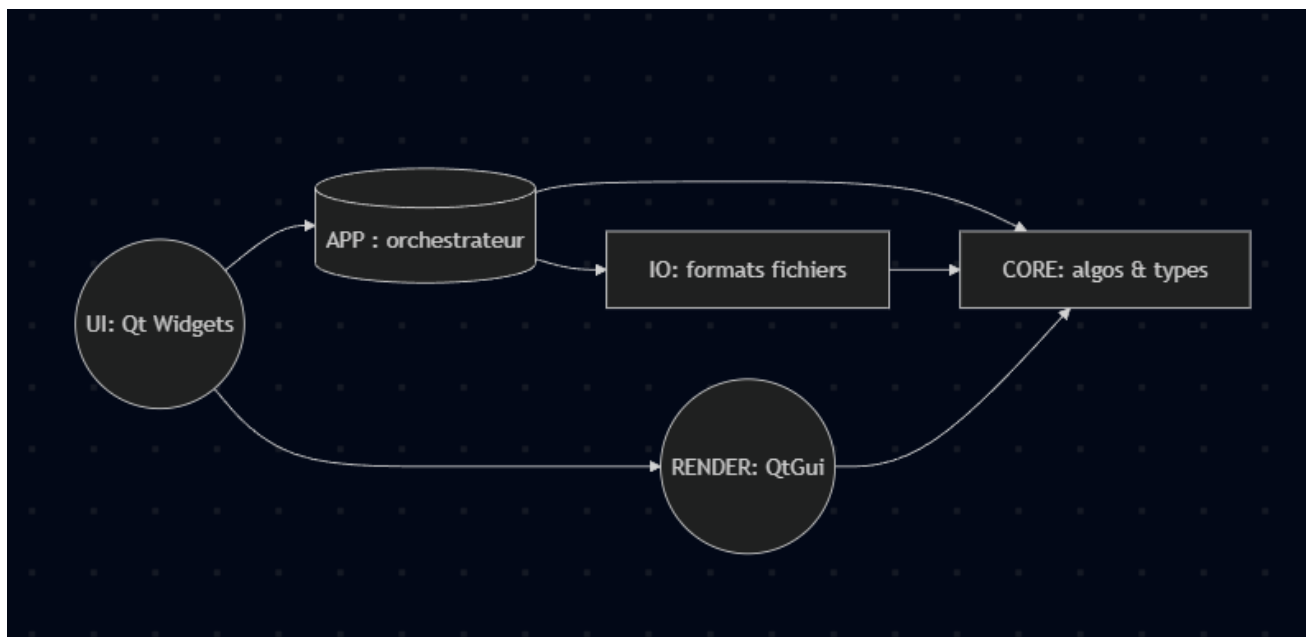
- Tests manuels d'acceptation passés
- Tests unitaires
- CI verte (build Win/Linux)
- Pas de régression de perf visible
- Documentation minimale (README et/ou guide utilisateur bref)

Architecture

Modèle utilisé : Architecture en couches

Objectif : Obtenir un couplage faible et une forte cohésion pour un logiciel maintenable, facilement testable et évolutif.

Vue d'ensemble



APP (C++) : propriétaire du Document*, annuler/restaurer, orchestration (*appelle CORE pour les traitements et IO pour l'ouverture/sauvegarde*).

UI (Qt Widgets) : interactions utilisateur et rafraîchissement de l'affichage.

CORE (C++) : types (*ImageBuffer, Layer, Selection, Document*) et algorithmes image (*pinceau, gomme, flood-fill, compositing/aplat*)

IO (C++) : ouverture/sauvegarde .epg et PNG/JPEG ; s'appuie sur CORE pour l'aplat

RENDER (QtGui) : composition vers QImage pour l'affichage, lecture seule du Document*.

*Document : structure **en mémoire** représentant l'image courante (calques, propriétés, sélection).

Règles de dépendances & responsabilités :

- ★ Mutations uniquement via APP ; CORE, IO, RENDER sont passifs.
- ★ RENDER n'a que le droit de lire le Document, ne le modifie pas.
- ★ IO ne dépend jamais de l'UI/Render.

Stratégie de performance et de rendu :

- Conversion **ImageBuffer** ↔ **QImage** pour l'affichage.
- Rendu **double-buffer**.
- Recomposition partielle via **ROI** (Region of Interest) pour la réactivité.
- Verrouillage minimal (éviter de bloquer le thread UI).

Src : [guideline_uml](#), [Types d'archi](#), [schema-architecture](#)

UML complet : [lucidchart](#)

Format de projet .epg

But

Le format .epg, permet de conserver un état non destructif du projet : calques, opacités, ordre, métadonnées — pour réouverture **à l'identique**.

Exactement comme GIMP a .xcf, nous avons décidé de créer notre propre format relatif au projet. Format custom minimaliste ainsi plus réaliste et contrôlable.

Conteneur

Fichier de type .epg

Encodage texte UTF8

Chemins : POSIX (/)

Arborescence exemple

```
project.epg
├─ meta.json
├─ preview.png
├─ doc/
│   └─ notes.md
└─ layers/
    ├── 0001.png
    ├── 0002.png
    └─ ...
```

```
"epg_version": 1,
"app": { "name": "EpiGimp2.0", "version": "1.0.0" },

"canvas": {
  "width": 1920,
  "height": 1080,
  "dpi": 72,
  "color_space": "sRGB",
  "background": { "r": 255, "g": 255, "b": 255, "a": 0 }
},

"layers": [
  {
    "id": "0001",
    "name": "test",
    "visible": true,
    "locked": false,
    "opacity": 1.0,
    "transform": {
      "tx": 0, "ty": 0, "scaleX": 1, "scaleY": 1, "rot": 0
    },
    "path": "layers/0001.png"
  }
],

"io": {
  "pixel_format_storage": "RGBA8_unorm_straight",
  "pixel_format_runtime": "ARGB32_premultiplied"
},

"misc": {
  "created_utc": "2025-10-22T21:45:00Z",
  "modified_utc": "2025-10-23T08:12:00Z"
}
```

Plan de livraison

Objectifs SMART (Specific, Mesurable, Achievable, Relevant and Timed)

Objectif	SPRINT (semaines)	Contenu	Critère mesurable	Preuves livrables
Setup & Cadrage	S1->S2	Mise en place de l'environnement de travail avec : Repertoire GitHub, CI minimale (build + tests vides), formatage (clang-format), lint (clang-tidy ruleset), choix techno, maquette UI, GitHub Project	Répertoire Github lié au Github Project Readme Cahier des charges CI Maquette validée	Lien repo Lien maquette UI
Avoir un logiciel exécutable sur Windows et Linux	S2 -> S4	Ouvrir PNG/JPEG, Créer projet, Enregistrer .epg, Exporter PNG/JPEG, Zoom/Pan, placeholders outils Avoir une V1 d'architecture	Fichier importer ouverture < 200ms Export PNG/JPEG OK Import invalides = erreur pas de crash	Le logiciel s'ouvre, importe un fichier, affiche et exporte ; erreurs gérées proprement UML
Edition de base	S4->S7	Développement des fonctionnalités basiques (pincesaux, gomme,	Latence de pinceau < 50ms Annuler Rétablir 20 steps sans	Interaction fluide, pas de lag visible, historique fiable, export

		selection, pot de peinture) Annuler Rétablir	pertes de données	conforme aux modifications
Calques	S7->S9	Gestion des calques (ordre, opacités et visibilité)		Calques fonctionnels, UX claire (preview + apply)
Stabilisation et performance	S9->S10	Amélioration des performances	CI OK Temps de démarrage < 1s	Binaire exécutable sur 2 OS, maintenance prête

Contraintes techniques

Catégorie	Description du défi	Solutions / pistes
Fichier	Possibilité d'ouvrir des fichiers images (png, jpeg, ...)	Fstream, QPixmap librairie's
UI	Afficher des images	Qt, QPixmap
Mémoire	Manipulation d'images haute résolution → consommation RAM élevée	Gestion des tiles / mipmaps, compression en mémoire
Annuler/Rétablir	Copier toute l'image à chaque étape est trop coûteux	Stocker des "diffs" (delta buffers) ou des opérations inverses
UI réactive	Ne pas bloquer le thread principal pendant le traitement	QtConcurrent / worker threads
Compatibilité	Multiplateforme (Linux, Windows)	Qt + CMake garantissent une bonne portabilité

Performanc e	Calculs pixel par pixel lents en CPU pur	SIMD (AVX2, SSE), GPU compute (OpenCL/Vulkan)
-------------------------	---	--

Risques identifiés et mesures

Risque	Description	Mesures / solutions
Fichiers image corrompus ou malicieux	Risque de crash ou corruption mémoire lors du parsing	- Validation stricte des headers - Utilisation de bibliothèques robustes (libpng, libjpeg-turbo) - Limitation de taille d'image (prévention OOM)
Perte de donnée lors d'un crash	Fermeture impromptue sans possibilité de sauvegarde	- Sauvegarde automatique et récupération après crash - Proposition de sauvegarde avant fermeture