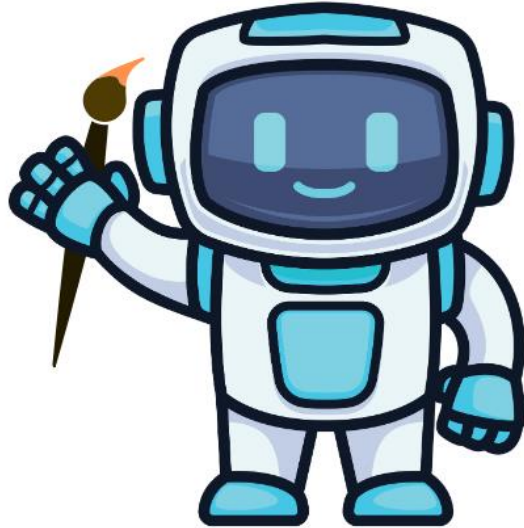


EpiGimp2.0



Cahier des charges & plan de livraison

Rédigé par : Apolline Fontaine et Léandre Godet

Pour : Epitech (client)

v1.1 : 22/10/2025

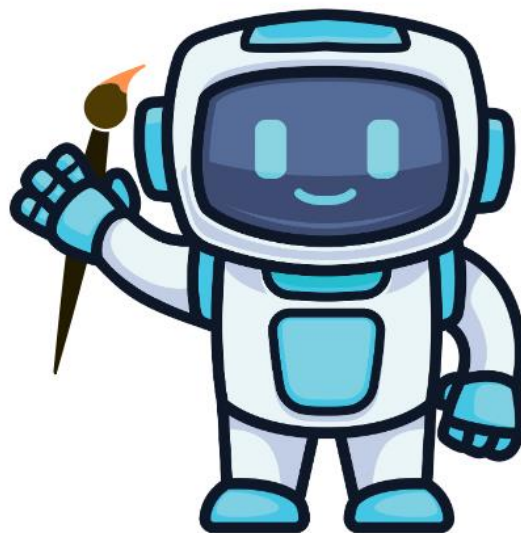
Sommaire

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

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 (7h) 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 (branches main, dev, PR obligatoires).

CI : GitHub Actions

Communication : Discord

Code qualité : Lint, Github Dependabot, libFuzzer

Technologies

C++20

CMake

QT 6.x

Périmètre

Must have

- Fichiers : Nouveau, Ouvrir PNG/JPEG, Enregistrer .epg (format natif remplaçant .xcf), Export PNG/JPEG.
- Calques : créer/supprimer/dupliquer, réordonner, visibilité, opacité
- Outils : Pinceau (couleur/dureté/taille), Gomme, Sélection rectangulaire, Pot de peinture.
- Historique : annuler/rétablir.

Should have

- Pipette, Texte,
- Raccourcis clavier,
- Zoom/Hand tool, grille/checker board transparence.

Could have

- Sélection libre (lasso),
- Courbes,
- Formats supplémentaires export WebP.

Exigences fonctionnelles

F-01 — Gestion des fichiers

Type : *Must have*

Description : Créer un nouveau projet, ouvrir des images (PNG/JPEG), enregistrer au format **.epg**, exporter en PNG/JPEG (image aplatie).

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.

F-02 — Calques

Type : *Must have*

Description : Créer / supprimer / dupliquer / verrouiller des calques, changer l'ordre (drag-and-drop), gérer la visibilité et l'opacité.

Critères d'acceptation

1. **Créer/Supprimer/Dupliquer/verrouiller** : les opérations apparaissent immédiatement dans la **pile des calques** et dans le rendu.
2. **Réordonner** : déplacer un calque modifie le rendu ;
3. **Visibilité** : activer/désactiver l'œil masque/affiche le calque instantanément.

4. **Opacité** : un slider 0–100 % affecte le rendu pixel-à-pixel.

F-03 — Outils d’édition (pinceau, gomme, sélection rectangulaire, pot de peinture)

Type : *Must have*

Description :

- **Pinceau** : taille, dureté, couleur (RGBA).
- **Gomme** : agit sur l’alpha du calque actif (ou du masque s’il est actif).
- **Sélection rectangulaire** : ajout/soustraction, bordure “marching ants”.
- **Pot de peinture** : remplissage zone connectée ou sélection, avec seuil de tolérance.

Critères d’acceptation

1. **Pinceau** : dessiner modifie uniquement la zone sous le trait ; Annuler restaure l’état précédent.
2. **Gomme** : rend transparent (alpha 0) les pixels gommés ; annuler restaure.
3. **Sélection** : la zone sélectionnée limite l’action des outils ; “Add/Subtract” modifie correctement le masque de sélection.
4. **Remplissage** : le seau affecte la zone attendue selon le seuil ; en présence d’une sélection, **seule** la sélection est remplie.

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

Type : *Must have*

Description : Empiler les actions en historique, permettre d’annuler/rétablir en cascade.

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).

F-05 — Pipette

Type : *Should have*

Description : Prélever la couleur d'un pixel du rendu (prend en compte l'aplat de tous les calques visibles).

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).

F-06 — Texte

Type : *Should have*

Description : Ajouter une zone de texte comme **nouveau calque** (police, taille, couleur, position).

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.

F-07 — Navigation & repères

Type : *Should have*

Description : Zoom, Pan, grille pour matérialiser la transparence.

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.

F-08 — Raccourcis clavier

Type : *Should have*

Description : Raccourcis communs basé sur ceux de GIMP :

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

F-09 — Sélection libre (lasso)

Type : *Could have*

Description : Tracer à main levée une sélection polygonale libre.

Critères d'acceptation

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

F-10 — Formats supplémentaires (export WebP)

Type : *Could have*

Description : Exporter en WebP (qualité paramétrable).

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.

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).
- **Flou gaussien** (rayon 5, 1080p) : **< 500 ms** avec aperçu réactif.
- **Fréquence de rendu** : **≥ 30 FPS** en édition courante (1080p, 1–3 calques).

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.

NF-03- Portabilité

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

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.

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.

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).

Performance & Benchmark

Élément mesuré	Méthode / outil	Cible
Ouverture PNG 1080p	QElapsedTimer ou std::chrono	< 200 ms sur images
Latence pinceau (moyenne 10 traits x 500px)	Chrono entre mousedown/up calcul moyen	< 50 ms
Allocation mémoire par image	valgrind –tool=massif ou heaptrack (local)	< 1,5× la taille de l'image brute
FPS UI pendant édition	Qt frame timing ou RenderDoc sur 5s	≥ 30 FPS
Temps de chargement/sauvegarde	Chronométrage automatisé	< 2 s pour images ≤ 20 Mo

Référence

- OS : Windows 11 23H2 (ou Ubuntu 24.04 LTS), alimentation “Performance”.
- CPU : Intel **Core i5-12400** (ou Ryzen 5 **5600G**).
- RAM : **16 Go** DDR4/DDR5 en dual-channel.
- GPU : **iGPU** (UHD 730 ou Vega 7).
- Stockage : NVMe (3 Go/s).
- Qt : **6.7.x**, Compiler : MSVC 19.3x (ou Clang 17/GCC 13).

Architecture & Modèle de données

Vue d'ensemble

- **UI (Qt Widgets)** : fenêtres, panneaux (Calques, Propriétés outil), canvas (QImage/QPixmap).
- **Core image** : buffers RGBA, outils (pinceau/gomme), sélection, filtres (OpenCV).
- **IO** : import PNG/JPEG, **format projet .epg**, export PNG/JPEG.
- **Historique** : pattern **Command** (do/annuler/rétablir).
- **Tâches** : **QtConcurrent / QThread** pour opérations coûteuses (flou, export).

Ouvrir/Créer → Pile de calques → Édition (outils) → Filtres (aperçu → appliquer) → Enregistrer (.epg) / Exporter (PNG/JPEG)

Rendu & performance

- Conversion **OpenCV Mat** ↔ **QImage** pour affichage.
- Rendu **double-buffer** ; prévisualisations via **ROI** (Region of Interest) pour réactivité.
- Verrouillage minimal (éviter de bloquer le thread UI).

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, Relevent and Timed)

Objectif	SPRINT (semaines)	Contenu	Critère mesurable	Preuves livrables
Setup & Cadrage	S1->S2	Mise en place de l'environnement de travail ave : 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
Avoir un logiciel exécutable sur Windows et Linux	S2 -> S4	Ouvrir PNG/JPEG, Créer projet, Enregistrer .eimg, Exporter PNG/JPEG, Zoom/Pan, placeholders outils	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
Edition de base	S4->S7	Développement des fonctionnalités basiques (pinceaux, gomme, selection, pot de	Latence de pinceau < 50ms Annuler Rétablir 20 steps sans	Interaction fluide, pas de lag visible, historique fiable, export conforme aux modifications

		peinture) Annuler Rétablir	pertes de données	
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, soutenance 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/Rét ablir	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
Compatibili té	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 imprévue sans possibilité de sauvegarde	- Sauvegarde automatique et récupération après crash - Proposition de sauvegarde avant fermeture

Lien maquette UI