

EPITA

Rapport de Soutenance

Projet S4 - Développement d'une application de peinture
numérique



FIGURE 1 – *
Logo du projet Rustique

Réalisé par :

Vincent NEDELIAN

Leonardo MONROC

Nabil BOROUS

Aurélien MAZE

Février 2025

Table des matières

1	Introduction	3
1.1	Présentation générale du projet	3
1.2	Objectifs du projet	3
1.3	Méthodologie de travail	4
2	État d'avancement du projet	5
2.1	Fonctionnalités implémentées	5
2.1.1	Structure de base et canvas	5
2.1.2	Outils de dessin	6
2.1.3	Gestion de fichiers	7
2.1.4	Interface utilisateur et navigation	7
2.2	Architecture technique	8
2.3	Tableau d'avancement des tâches	9
3	Répartition des tâches et contributions individuelles	11
3.1	Vincent Nedelian - Outils de dessin et palette de couleurs	11
3.1.1	Mes responsabilités	11
3.1.2	Réalisations et défis	11
3.1.3	Perspectives	12
3.2	Leonardo Monroc - Canvas et couleurs	12
3.2.1	Mes responsabilités	12
3.2.2	Développement et optimisations	12
3.2.3	Perspectives pour les calques	13
3.3	Nabil Borous - Gestion de fichiers et zoom	13
3.3.1	Mes responsabilités	13
3.3.2	Systèmes implémentés	13
3.3.3	Perspectives	13
3.4	Aurélien Maze - Interface utilisateur et site web	14
3.4.1	Mes responsabilités	14
3.4.2	Interface développée	14
3.4.3	Perspectives	14
4	Comparaison avec le planning initial	15
4.1	Avancées et retards	15
4.1.1	Avancées	15

4.1.2	Retards	15
4.2	Ajustements effectués	15
5	Plan de travail pour la prochaine période	16
5.1	Fonctionnalités prévues	16
5.1.1	Système de calques	16
5.1.2	Outils avancés	16
5.1.3	Gestion avancée des fichiers	16
5.1.4	Améliorations de l'interface	16
5.2	Répartition des tâches futures	17
5.3	Calendrier prévisionnel	17
6	Défis techniques et solutions	18
6.1	Performance du rendu	18
6.1.1	Défi	18
6.1.2	Solution	18
6.2	Gestion de la transparence	18
6.2.1	Défi	18
6.2.2	Solution	18
6.3	Algorithme de remplissage	18
6.3.1	Défi	18
6.3.2	Solution	19
6.4	Zoom et pan fluides	19
6.4.1	Défi	19
6.4.2	Solution	19
7	Conclusion	20
7.1	Bilan de l'avancement	20
7.2	Apports personnels et professionnels	20
7.3	Perspectives	20

1

Introduction

1.1 Présentation générale du projet

Notre projet *Rustique* vise à développer une application de peinture numérique en Rust. Ce logiciel a pour objectif d’offrir un outil de dessin fonctionnel tout en nous permettant d’explorer les fondamentaux de la programmation en Rust et du développement d’interfaces graphiques.

En tant que responsable de la présentation générale du projet, j’ai (Nabil Borous) coordonné notre vision pour *Rustique*. Notre équipe a choisi ce projet pour plusieurs raisons :

- Développer nos compétences en Rust, un langage réputé pour sa sécurité et ses performances. Rust offre des garanties de sécurité mémoire sans ramasse-miettes, ce qui en fait un choix idéal pour des applications performantes.
- Explorer les défis techniques liés au traitement d’images et au rendu graphique. Ce domaine complexe nécessite à la fois performances et précision, correspondant parfaitement aux forces de Rust.
- Créer un logiciel utile avec une interface intuitive. Nous voulions que notre projet ait une application concrète et puisse servir à des utilisateurs réels.
- Apprendre à travailler en équipe sur un projet complexe, nous donnant une expérience proche des conditions réelles de développement.

Notre application permettra aux utilisateurs de créer des œuvres d’art numériques à l’aide de divers outils. Ces applications sont utilisées dans l’art numérique, le design graphique et l’illustration. Notre objectif est de fournir une interface intuitive pour la création artistique tout en servant de plateforme d’apprentissage.

1.2 Objectifs du projet

Nos objectifs initiaux, définis dans le cahier des charges, sont les suivants :

1. Créer une application de peinture numérique fonctionnelle et extensible, capable

de répondre aux besoins de base des utilisateurs tout en permettant d'ajouter de nouvelles fonctionnalités.

2. Apprendre à utiliser Rust dans un contexte de développement d'applications graphiques, en explorant ses capacités pour la gestion de données (pixels) et les interactions en temps réel.
3. Implémenter des fonctionnalités de base de dessin, de manipulation d'images et de gestion de fichiers.
4. Développer des compétences en conception d'interfaces utilisateur et en traitement d'images.
5. Mettre en place une architecture modulaire qui facilitera l'ajout de nouvelles fonctionnalités.

Ces objectifs guident notre travail et restent nos priorités pour les développements futurs.

1.3 Méthodologie de travail

Notre équipe a adopté une approche agile pour le développement. Cette méthodologie nous permet d'être flexibles et de nous adapter rapidement aux défis. Nous avons divisé le projet en modules fonctionnels et attribué des responsabilités spécifiques à chaque membre.

Pour le suivi du projet, nous utilisons un tableau Kanban qui nous permet de visualiser l'état d'avancement de chaque fonctionnalité. Les tâches passent par différentes étapes : "À faire", "En cours", "En test" et "Terminé".

Pour la gestion du code, nous utilisons Git avec un dépôt GitHub, permettant un travail simultané sur différentes parties du projet. Nous avons mis en place des revues de code obligatoires avant l'intégration dans la branche principale.

Nous organisons deux réunions d'équipe hebdomadaires : planification en début de semaine et rétrospective en fin de semaine. Un canal de communication dédié permet les échanges quotidiens et les questions urgentes.

La documentation est maintenue au fur et à mesure du développement, facilitant la collaboration et garantissant la maintenabilité du code.

2

État d'avancement du projet

2.1 Fonctionnalités implémentées

2.1.1 Structure de base et canvas

Le cœur de notre application repose sur un système de canvas robuste qui offre :

- **Création de toile personnalisable** : Dimensions spécifiques jusqu'à 4000×4000 pixels.
- **Gestion des pixels** : Manipulation individuelle avec support complet de la transparence.
- **Rendu optimisé** : Système qui ne rafraîchit que les parties modifiées du canvas pour des performances fluides.
- **Visualisation de la transparence** : Motif en damier sous les zones transparentes pour distinguer clairement les parties visibles et invisibles.

La structure utilise un vecteur unidimensionnel avec conversion d'indices 2D en 1D pour optimiser les performances.

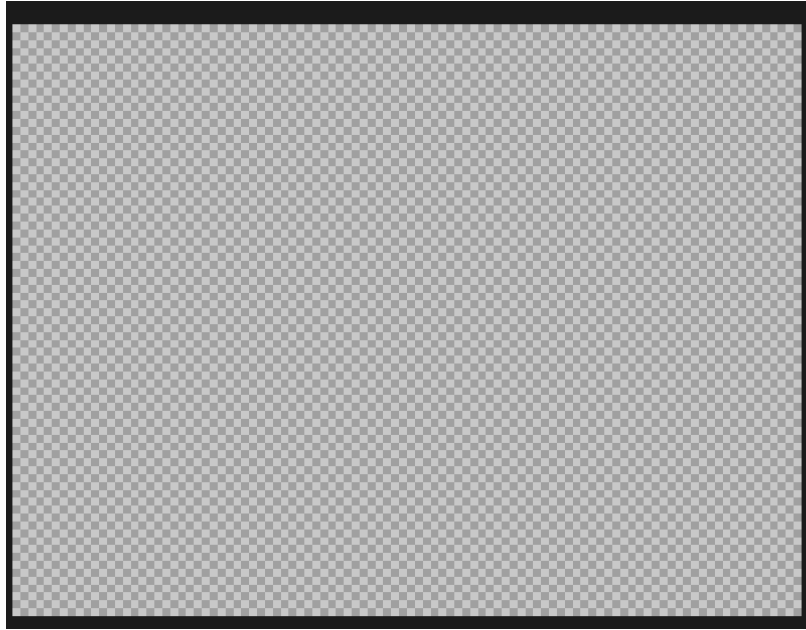


FIGURE 2.1 – Le canvas de Rustique avec affichage en damier pour la transparence

2.1.2 Outils de dessin

Nous avons implémenté quatre outils fondamentaux :

- **Pinceau** : Dessin à main levée avec couleur et taille ajustables. Forme circulaire avec interpolation entre points pour des traits fluides.
- **Gomme** : Efface les pixels en les rendant transparents, avec une taille ajustable indépendamment du pinceau.
- **Seau de remplissage** : Remplit une zone continue de pixels de même couleur, utilisant un algorithme optimisé pour éviter les dépassements de pile.
- **Pipette** : Prélève une couleur existante sur le canvas pour l'utiliser avec d'autres outils.

Ces outils utilisent des algorithmes optimisés pour garantir une expérience fluide, même sur des images de grande taille.

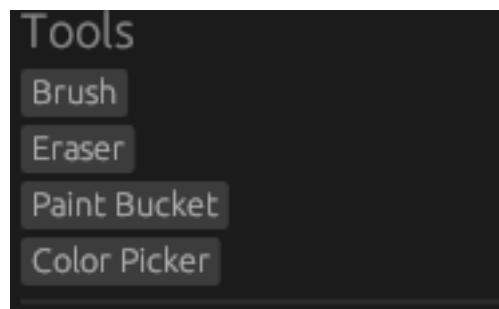


FIGURE 2.2 – Interface montrant les différents outils de dessin disponibles

2.1.3 Gestion de fichiers

Notre implémentation actuelle permet de :

- **Sauvegarder au format PNG** : Support natif de la transparence et compression sans perte.
- **Utiliser des dialogues natifs** : Interface cohérente avec le système d'exploitation pour une expérience familière.
- **Gérer les extensions** : Ajout automatique de l'extension .png si nécessaire.
- **Détecter les modifications** : Alerte l'utilisateur si des changements n'ont pas été sauvegardés.

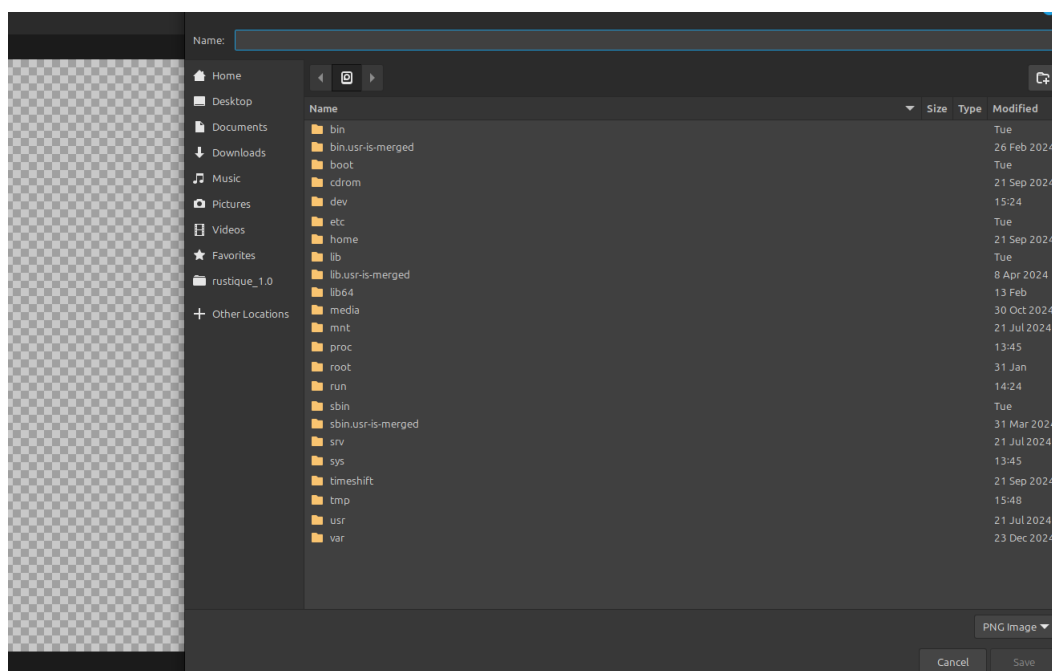


FIGURE 2.3 – Dialogue de sauvegarde de fichier dans Rustique

2.1.4 Interface utilisateur et navigation

Notre interface comprend :

- **Panneau latéral** : Regroupe tous les outils et options, organisés logiquement.
- **Sélecteur de couleur** : Composant intuitif avec support RVB et alpha.
- **Contrôles paramétriques** : Curseurs et champs pour ajuster finement les paramètres des outils.
- **Zoom avancé** : Centré sur le curseur, allant de 10
- **Déplacement du canvas** : Navigation fluide avec le bouton central de la souris.

- **Affichage adaptatif** : S'ajuste automatiquement à la taille de fenêtre.

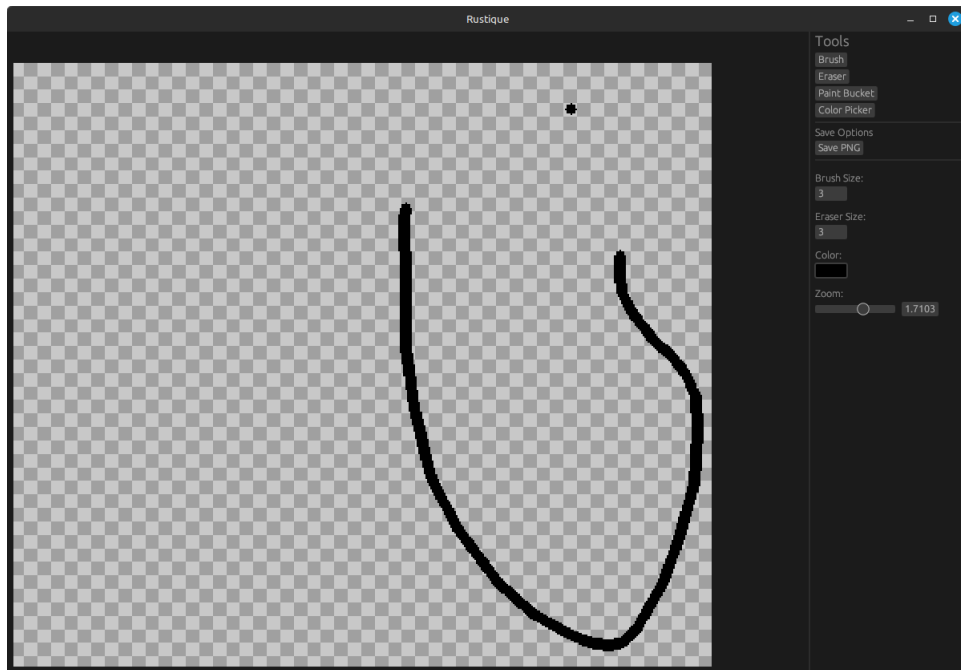


FIGURE 2.4 – Vue d'ensemble de l'interface utilisateur avec zoom activé

2.2 Architecture technique

Notre application est organisée en modules distincts :

- **Module Canvas (canvas.rs)** : Gère la structure de données du canvas et les opérations sur les pixels.
- **Module Paint (paint.rs)** : Implémente les outils de dessin et la logique principale de l'application.
- **Module Main (main.rs)** : Point d'entrée de l'application et configuration initiale.

Cette organisation modulaire facilite le développement collaboratif et les extensions futures.

Pour le développement, nous utilisons :

- **eframe/egui** : Framework d'interface graphique léger et moderne pour Rust.
- **image** : Bibliothèque pour la manipulation et l'enregistrement d'images.
- **rfd** : Bibliothèque pour les dialogues de fichiers multi-plateformes.

Pourquoi egui/eframe ?

Nous avons choisi egui pour sa simplicité d'utilisation, son intégration native avec Rust et ses performances. Contrairement à d'autres bibliothèques, egui ne nécessite aucun fichier de configuration externe et fonctionne identiquement sur toutes les plateformes.

Sa nature réactive et sa mise à jour en temps réel en font un choix parfait pour une application de dessin qui nécessite un retour visuel immédiat. L'approche "immediate mode" simplifie le développement d'interfaces dynamiques où l'état change fréquemment.

2.3 Tableau d'avancement des tâches

Responsable	Tâche	Février 2025	Avril 2025	Mai 2025
4*Vincent	Pinceau	90%	100%	100%
	Gomme	90%	100%	100%
	Seau de remplissage	80%	100%	100%
	Formes géométriques	0%	60%	100%
4*Leonardo	Structure Canvas	95%	100%	100%
	Rendu graphique	85%	95%	100%
	Système de calques	0%	70%	95%
	Modes de fusion	0%	40%	80%
4*Nabil	Sauvegarde PNG	85%	100%	100%
	Chargement d'images	10%	80%	100%
	Système de zoom	80%	95%	100%
	Format personnalisé	0%	50%	90%
4*Aurélien	Interface de base	90%	100%	100%
	Interface de calques	0%	70%	95%
	Raccourcis clavier	5%	80%	100%
	Site web du projet	70%	90%	100%

TABLE 2.1 – Avancement détaillé des tâches par membre et par période

Module	Février 2025	Avril 2025	Mai 2025
Outils de base	85%	100%	100%
Canvas et rendu	90%	98%	100%
Gestion de fichiers	50%	90%	100%
Interface utilisateur	80%	95%	100%
Système de calques	0%	70%	95%
Outils avancés	0%	60%	90%

TABLE 2.2 – Avancement global par module et par période

Phase du projet	Tâches accomplies	Tâches planifiées
Février 2025	<ul style="list-style-type: none"> — Implémentation des outils de base — Structure du canvas — Interface utilisateur — Système de zoom — Sauvegarde PNG 	<ul style="list-style-type: none"> — Finalisation du système de navigation — Optimisation des performances — Documentation utilisateur initiale
Avril 2025	<ul style="list-style-type: none"> — Système de calques (base) — Chargement d'images — Premières formes géométriques — Raccourcis clavier 	<ul style="list-style-type: none"> — Modes de fusion des calques — Format de fichier personnalisé — Outils de sélection
Mai 2025	<ul style="list-style-type: none"> — Finalisation des calques — Modes de fusion complets — Ensemble d'outils géométriques — Format de fichier avec calques 	

TABLE 2.3 – Répartition des tâches par phase du projet

3

Répartition des tâches et contributions individuelles

3.1 Vincent Nedelian - Outils de dessin et palette de couleurs

3.1.1 Mes responsabilités

En tant que responsable des outils de dessin, j'ai (Vincent) été chargé de :

- Concevoir et implémenter les outils de base (pinceau, gomme, seau, pipette)
- Développer les algorithmes nécessaires pour ces outils
- Gérer les paramètres des outils (taille, opacité)
- Intégrer la sélection et l'application des couleurs

3.1.2 Réalisations et défis

J'ai implémenté le pinceau, outil fondamental qui applique une couleur avec une forme circulaire de taille variable. Pour garantir un dessin fluide lors des mouvements rapides, j'ai développé un algorithme qui interpole les points entre deux positions successives de la souris.

Pour l'outil gomme, j'ai réutilisé la logique du pinceau en adaptant le système pour rendre les pixels transparents. Cela a nécessité une collaboration avec Leonardo pour la gestion de la transparence.

L'implémentation du seau de remplissage a été particulièrement intéressante. L'algorithme classique utilise une approche récursive, mais celle-ci peut provoquer des dépassements de pile pour les grandes zones. J'ai donc opté pour une implémentation itérative avec une file d'attente, plus robuste et efficace.

Un défi majeur concernait l'optimisation des performances pour les pinceaux de grande taille. J'ai résolu ce problème en pré-calculant les offsets des pixels, améliorant significativement les performances.

Défi des grands pinceaux

Les pinceaux de grande taille (>30 pixels de rayon) causaient des ralentissements. L'analyse a révélé que le calcul répété de la distance au centre pour chaque pixel était coûteux.

Ma solution a été de pré-calculer les offsets des pixels affectés, évitant les calculs redondants. Cette optimisation a amélioré les performances d'un facteur 5 pour les grands pinceaux.

3.1.3 Perspectives

Pour les prochaines étapes, je compte ajouter :

- Différents styles de pinceau (doux, texturé)
- Mode aérographe avec contrôle de l'opacité
- Outils de formes géométriques
- Sélection et manipulation de régions

3.2 Leonardo Monroc - Canvas et couleurs

3.2.1 Mes responsabilités

En tant que responsable du système de canvas, j'ai (Leonardo) été chargé de :

- Concevoir la structure de données pour le canvas
- Développer le système de rendu graphique
- Gérer l'interaction canvas-outils
- Implémenter le support de la transparence

3.2.2 Développement et optimisations

J'ai développé une structure de données efficace utilisant un vecteur unidimensionnel avec des valeurs représentant soit une couleur, soit la transparence. Cette approche permet un accès rapide aux pixels et une gestion élégante de la transparence.

Pour le rendu, j'ai implémenté un système qui convertit notre structure en texture utilisable par egui, avec un mécanisme de "dirty flag" qui n'actualise la texture que lorsque nécessaire, optimisant ainsi les performances.

La visualisation des zones transparentes est assurée par un motif en damier généré dynamiquement lors du rendu, ce qui aide les utilisateurs à distinguer clairement les

zones transparentes de leur image.

3.2.3 Perspectives pour les calques

Pour la prochaine phase, je prévois d'implémenter :

- Une structure pour les calques multiples
- Des opérations de base (création, suppression, réorganisation)
- Des modes de fusion entre calques
- L'intégration avec les outils existants

3.3 Nabil Borous - Gestion de fichiers et zoom

3.3.1 Mes responsabilités

En tant que responsable de la gestion des fichiers et du zoom, j'ai (Nabil) été chargé de :

- Implémenter la sauvegarde et le chargement des fichiers
- Développer le système de zoom et de navigation
- Gérer les formats d'image
- Superviser la coordination générale

3.3.2 Systèmes implémentés

J'ai implémenté la sauvegarde au format PNG avec support de la transparence, en utilisant les bibliothèques `image` et `rfd` pour une interface native. Le système ajoute automatiquement l'extension `.png` si nécessaire.

Pour la navigation, j'ai développé un système de zoom centré sur le curseur et un déplacement fluide avec le bouton central de la souris, offrant une expérience intuitive même sur de grandes images.

3.3.3 Perspectives

Pour la prochaine phase, je prévois de :

- Implémenter le chargement d'images dans différents formats
- Développer un format de fichier personnalisé pour les projets avec calques
- Ajouter des options d'exportation avancées

- Optimiser davantage les performances

3.4 Aurélien Maze - Interface utilisateur et site web

3.4.1 Mes responsabilités

En tant que responsable de l'interface utilisateur, j'ai (Aurélien) été chargé de :

- Concevoir l'interface utilisateur
- Développer les interactions (menus, boutons)
- Créer une expérience utilisateur intuitive
- Développer le site web de présentation

3.4.2 Interface développée

J'ai conçu une interface organisée autour d'un panneau latéral pour les outils et d'une zone centrale pour le canvas. L'interface inclut des boutons pour les outils, des contrôles pour les paramètres et un sélecteur de couleur, le tout réactif aux modifications en temps réel.

Un aspect important a été la gestion des interactions utilisateur-canvas, incluant la détection des clics et mouvements, et leur conversion en coordonnées canvas avec prise en compte du zoom et du déplacement.

J'ai également créé un site web de présentation avec un design responsive, des captures d'écran des fonctionnalités et une section dédiée à l'équipe.

3.4.3 Perspectives

Pour la prochaine phase, je prévois d'ajouter :

- Une interface de gestion des calques
- Des raccourcis clavier
- Un historique des actions (annuler/refaire)
- Des thèmes clairs/sombres

4

Comparaison avec le planning initial

4.1 Avancées et retards

4.1.1 Avancées

Nous avons progressé significativement sur plusieurs aspects :

- **Outils de base** : Implémentation complétée selon le planning
- **Interface utilisateur** : Développement terminé avec avance
- **Sauvegarde d'images** : Fonctionnalité opérationnelle
- **Système de zoom** : Implémenté avec avance sur le planning

4.1.2 Retards

Certaines fonctionnalités ont pris du retard :

- **Système de calques** : Reporté à la prochaine étape
- **Chargement d'images** : Pas encore complètement implémenté
- **Formes géométriques** : Reporté à la prochaine phase

4.2 Ajustements effectués

Suite à l'analyse de notre progression, nous avons :

- Priorisé les fonctionnalités de base et l'expérience utilisateur
- Consacré plus de temps à l'optimisation des performances
- Réaffecté certaines ressources pour accélérer le développement

Ces ajustements nous ont permis de maintenir un rythme soutenu malgré les retards sur certaines fonctionnalités.

5

Plan de travail pour la prochaine période

5.1 Fonctionnalités prévues

5.1.1 Système de calques

- Création et gestion de calques multiples
- Contrôle de visibilité et d'opacité
- Réorganisation et fusion de calques
- Modes de fusion

5.1.2 Outils avancés

- Formes géométriques (rectangles, ellipses, lignes)
- Outil de sélection
- Styles de pinceau avancés
- Outil de texte

5.1.3 Gestion avancée des fichiers

- Chargement d'images multi-formats
- Format de fichier personnalisé pour les projets
- Options d'exportation avancées

5.1.4 Améliorations de l'interface

- Interface de gestion des calques
- Raccourcis clavier

- Historique des actions
- Thèmes clairs/sombres

5.2 Répartition des tâches futures

Chaque membre continuera à développer son domaine de responsabilité :

- **Vincent** : Outils de formes géométriques, styles de pinceau avancés
- **Leonardo** : Système de calques, modes de fusion
- **Nabil** : Chargement d'images, format de fichier personnalisé
- **Aurélien** : Interface de gestion des calques, raccourcis, historique

5.3 Calendrier prévisionnel

- **Mars 2025 (Semaines 1-2)** : Début implémentation calques, amélioration interface
- **Mars 2025 (Semaines 3-4)** : Finalisation calques de base, début outils avancés
- **Avril 2025 (Semaines 1-2)** : Modes de fusion, formes géométriques, format personnalisé
- **Avril 2025 (Semaines 3-4)** : Tests, corrections, optimisations

6

Défis techniques et solutions

6.1 Performance du rendu

6.1.1 Défi

Assurer des performances fluides avec grands canvas et pinceaux de grande taille.

6.1.2 Solution

- Système de dirty flag pour éviter les rendus inutiles
- Optimisation des algorithmes de dessin
- Mise en cache des calculs coûteux

6.2 Gestion de la transparence

6.2.1 Défi

Distinguer entre pixels non définis et pixels avec transparence.

6.2.2 Solution

- Représentation claire des pixels transparents vs. définis
- Motif en damier pour visualiser les zones transparentes

6.3 Algorithme de remplissage

6.3.1 Défi

Éviter les dépassements de pile avec l'approche récursive traditionnelle.

6.3.2 Solution

- Implémentation itérative avec file d'attente
- Tableau de visite pour éviter les traitements multiples

6.4 Zoom et pan fluides

6.4.1 Défi

Créer un système de navigation intuitif et performant.

6.4.2 Solution

- Zoom centré sur le curseur
- Système de transformation de coordonnées précis

7

Conclusion

7.1 Bilan de l'avancement

À ce stade, nous avons posé des bases solides pour notre application avec :

- Un canvas performant avec support de la transparence
- Des outils de dessin de base fonctionnels
- Une sauvegarde d'images opérationnelle
- Une interface intuitive
- Un système de navigation fluide

7.2 Apports personnels et professionnels

Ce projet nous a permis d'acquérir de nouvelles compétences :

- **Techniques** : Maîtrise de Rust, interfaces graphiques, traitement d'images
- **Gestion** : Planification, suivi, adaptation
- **Collaboration** : Communication, travail d'équipe, partage de connaissances

7.3 Perspectives

Pour la suite, nous sommes enthousiastes à l'idée d'implémenter des fonctionnalités avancées comme les calques et les formes géométriques.

À plus long terme, nous envisageons :

- Un mode de dessin collaboratif
- Des filtres et effets inspirés du thème de la rouille
- Un système de plugins extensible
- Des versions pour d'autres plateformes

Bien que nous n'ayons pas implémenter toutes les fonctionnalités voulues pour cette première soutenance, nous sommes satisfaits de notre avancement et confiants dans notre capacité à mener ce projet à terme avec succès.